

(11)特許出願公開番号

特開平11-265395

(43)公開日 平成11年(1999)9月28日

(51) Int.Cl.⁶

識別記号

FI

G O 6 F 17/30
12/00

5 1 3

G O 6 F 15/40
12/00
15/40

3 8 0 D
5 1 3 D
3 4 0 D

審査請求 未請求 請求項の数 9 OL (全 19 頁)

(21)出願番号 特願平10-68099

(22)出題日 平成10年(1998)3月18日

(71)出願人 000005108

株式会社日立製作所

東京都千代田区神田駿河台四丁目6番地

(72) 發明者 鈴木 豊人

神奈川 神奈川県川崎市幸区鹿島田890番地 株式会社日立製作所情報・通信開発本部内

(72)發明者 小池 博

神奈川 川崎市幸区鹿島田890番地 株式
会社日立製作所情報・通信開発本部内

(72)発明者 溝手 裕二

神奈川県川崎市幸区鹿島田890番地 株式会社日立製作所情報・通信開発本部内

(74) 代理人 弁理士 小川 勝男

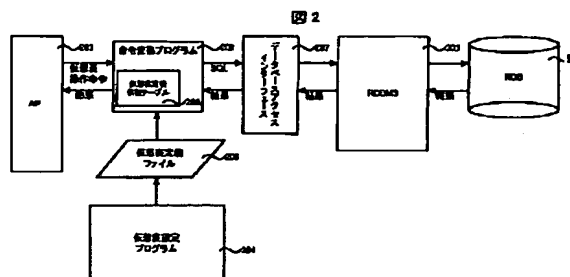
最終頁に続く

(54) 【発明の名称】 データベースアクセスシステム

(57) 【要約】

【課題】本発明は、アプリケーションの実行者が、アプリケーションの対象テーブルを、データベース管理システムの定義を変更することなく、変更することを可能とする。

【解決手段】データベースの1つ以上の実テーブルの実カラムデータに対して操作を行う1つ以上のSQL文を生成し、該SQL文を、該アクセス制御手段に与えることにより、実カラムデータの取得または変更を操作し、該実カラムデータの取得または変更の結果を、仮想表カラムデータの取得または変更結果として出力するデータベース操作命令変換処理プログラムを持つことにある。



【特許請求の範囲】

【請求項1】1種類以上の実カラムデータを有する実レコードの集合である実テーブルを、複数備えているデータベースと、該データベースの実カラムデータを取得または変更するデータベース管理手段と、与えられたSQL文に対応して、前記データベース管理手段の該実カラムデータの取得または変更動作を制御するアクセス制御手段からなるデータベースアクセスシステムにおいて、該データベースの1つ以上の実テーブルのいくつかの実カラムデータを、1つの仮想表の各仮想表カラムデータとして関連付ける仮想表定義情報と、該仮想表カラムデータを取得または変更する仮想表操作命令を入力とし、該データベースの1つ以上の実テーブルの実カラムデータに対して操作を行う1つ以上のSQL文を生成し、該SQL文を、該アクセス制御手段に与えることにより、実カラムデータの取得または変更を操作し、該実カラムデータの取得または変更の結果を、仮想表カラムデータの取得または変更結果として出力する、データベース操作命令変換処理プログラムを持つことを特徴とするデータベースアクセスシステム。

【請求項2】1種類以上の実カラムデータを有する実レコードの集合である実テーブルを、複数備えているデータベースと、該データベースの実カラムデータを取得または変更するデータベース管理手段と、与えられたSQL文に対応して、前記データベース管理手段の該実カラムデータの取得または変更動作を制御するアクセス制御手段を対象とし、該データベースの1つ以上の実テーブルのいくつかの実カラムデータを、1つの仮想表の各仮想表カラムデータとして仮想化する仮想表定義情報と、該仮想表カラムデータを取得または変更する仮想表操作命令を入力とし、該データベースの1つ以上の実テーブルの実カラムデータに対して操作を行う1つ以上のSQL文を生成し、該SQL文を、該アクセス制御手段に与えることにより、実カラムデータの取得または変更を操作し、該実カラムデータの取得または変更の結果を、仮想表カラムデータの取得または変更結果として出力するデータベース操作命令変換処理プログラム。

【請求項3】請求項1又は2記載のデータベース操作命令変換処理プログラムにおいて、前記仮想表定義情報は、仮想表カラム名と実テーブルカラム名を関連付ける対照情報と、該仮想表を構成する1つ以上の実テーブルを関連付ける実カラム名の集合情報から構成することを特徴とするデータベース操作命令変換処理プログラム及び、データベースアクセスシステム。

【請求項4】請求項1又は2記載において、前記データベース操作命令変換処理プログラムへの入力となり、データベースの1つ以上の実テーブルのいくつかの実カラムデータを、1つの仮想表の各仮想表カラムデータとして関連付け、仮想表を構成するための、仮想表カラム名と実テーブルカラム名を関連付ける対照情報と、該仮想

表を構成する1つ以上の実テーブルを関連付ける実カラム名の集合からなる、仮想表定義情報を記録したことを特徴とした媒体。

【請求項5】請求項1又は2記載において、仮想表定義情報を記録した媒体の識別名称を、前記データベース操作命令変換処理プログラムの起動命令に与え、起動することにより、該データベース操作命令変換処理プログラムが対象とする仮想表を決定することを特徴とするデータベースアクセスシステム。

【請求項6】請求項1又は2記載において、前記仮想表に対する仮想表レコード検索操作命令は、検索対象となる仮想表カラム名列と仮想表レコード検索条件の入力から、該仮想表定義情報と、該入力情報中に出現する仮想表カラム名から、検索対象とする実テーブル名を解決し、該検索対象仮想表カラム名列中に出現する仮想表カラム名を実テーブルカラム名に変換して検索対象実テーブルカラム名列とし、該仮想表レコード検索条件中に出現する仮想表カラム名を実テーブルカラム名に変換して実テーブルレコード検索条件とし、該仮想表定義情報と、該検索対象とする実テーブル名より、テーブル結合条件を生成し、該検索対象実テーブルカラム名列と、該実テーブルレコード検索条件と、該検索対象とする実テーブル名と、該実テーブル結合条件から、SQLのSELECT文を生成し、該データベースのアクセス制御手段を用いて該生成したSQLのSELECT文を発行し、該検索結果のカラム名を、仮想表カラム名に変換して返すことによって仮想表レコードの検索を行うデータベース操作命令変換プログラムを持つことを特徴とするデータベースアクセスシステム。

【請求項7】請求項1又は2記載において、前記仮想表に対する仮想表レコード挿入操作命令は、前記仮想表定義情報と、仮想表カラム名と仮想表カラムデータの対の列からなる挿入仮想表レコード情報の入力から、該挿入仮想表レコード情報を各挿入対象実テーブル単位に分割し、該実テーブル単位挿入レコードの各仮想表カラム名を実テーブルカラム名に変換し、各挿入対照の候補となる実テーブルに対し、整合性確認のSQLのSELECT文を生成して仮想表レコード挿入後の整合性の確認を行い、整合性が保たれている場合、各実テーブル毎のSQLのINSERT文を生成し、該データベースのアクセス制御手段を用いて該生成したSQLのINSERT文を発行することによって、該仮想表レコードの挿入を行うデータベース操作命令変換プログラムを持つことを特徴とするデータベースアクセスシステム。

【請求項8】請求項1又は2記載において、前記仮想表に対する仮想表レコード削除操作命令は、前記仮想表定義情報と、仮想表レコード削除条件の入力から、仮想表レコード削除条件中の仮想表カラム名を実テーブルカラム名に変換して、対象となる実テーブル毎に分割し、該仮想表を構成する実テーブルの全ての組に関して削除後

のデータ整合性が保証できる実テーブルレコードの削除条件を生成し、該実テーブルレコード削除条件をもとに、各テーブル毎のSQLのDELETE文を生成し、該データベースのアクセス制御手段を用いて該生成したSQLのDELETE文を発行することによって、仮想表レコードの削除を行うデータベース操作命令変換プログラムを持つことを特徴とするデータベースアクセスシステム。

【請求項9】請求項1又は2記載において、前記仮想表に対する仮想表レコード更新操作命令は、前記仮想表定義情報と、仮想表カラム名と仮想表カラムデータの組からなる仮想表更新レコード情報と、仮想表レコード更新条件から、仮想表全カラム名を検索対象仮想表カラム名、該仮想表レコード更新条件を仮想表レコード検索条件入力とする前記仮想表レコード検索操作を行い、該検索結果仮想表レコードの該仮想表更新レコード情報に含まれる仮想表カラムのデータを該仮想表更新レコード情報の仮想表カラムデータで置換し、該仮想表レコード更新条件を、仮想表レコード削除条件入力とする前記仮想表レコード削除操作を行い、該置換後検索結果仮想表レコードを入力として、前記仮想表レコード挿入操作を行うことによって、仮想表レコードの更新操作を行うデータベース操作命令変換プログラムを持つことを特徴とするデータベースアクセスシステム。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、データベースのレコードを使用するアプリケーションプログラムに係り、特にアクセス対象とするテーブル構造の変更に強いアクセスプログラムに関する。

【0002】

【従来の技術】特開平9-6801号公報（データベースアクセスシステム）によれば、近年、様々な種類のデータを共有し、情報の有効活用を図るためのデータベース管理システムが、提案されている。現在提案されている各種のデータベース管理システムの中でも、データの関連付けを考慮したテーブルを利用してデータを取り扱う、リレーショナルデータベース管理システムが急速に普及している。

【0003】リレーショナルデータベースでのデータ記憶の基本的な考え方は、テーブルを利用することによって行なう。テーブルは、カラム（列）とレコード（行）からなる二次元の表であり、表を構成する各要素は、カラムデータと称されている。また、通常、カラム（列）は複数存在するので、1行、即ち、1レコードは、複数種類のカラムデータによって構成されている。

【0004】このようなリレーショナルデータベースの管理システムは、今日のビジネス社会に不可欠なシステムとなっており、ISO（国際標準化機構）は、リレーショナルデータベースを定義・操作するために、リレー

ショナルデータベース用言語である「SQL（Structured Query Language）構文」を規定している。

【0005】「HiRDB SQLリファレンス」（株式会社 日立製作所）によれば、データベース管理システムでは、「ビュー」機能を提供している。ビューは、別のテーブル等からデータをアクセスすることができる論理表である。実際には、ビューにデータそのものが格納されているわけではなく、ビューの基盤となっているテーブルをビューに対して、「実表」と呼ぶ。

10 【0006】データベースアプリケーションシステム開発時、ビューを使用する利点としては、実際には複数の実表を基にして構成されているデータであっても、1つのテーブルのデータとしてアクセスすることができ、実際のテーブル構造の複雑性を意識する必要が無い。実際には実表の定義を変更することなく列を改名することによって別の視点から見たデータを提示することができる等が挙げられる。

20 【0007】ビューはSQLのSELECT文によって定義され、SELECT文の検索実行結果レコードの集合がそのまま一つのビューとなり、ビューのカラム名はビューの定義時指定された名前となる。

【0008】アプリケーションシステムの操作対象としてビューを使用した場合、「特開平9-237280号公報」（ERモデル指向によるデータベース検索装置）によれば、テーブルの拡張や、アプリケーション内でのカラムデータの意味付けの変更などの際、データベース管理システムのビュー定義を直接変更することによって対応する必要があり、それはアプリケーションプログラマがデータベース管理者に依頼して行う作業である。

30 【0009】また、ビューを定義のSELECT文が「結合」の条件を含む場合、更新操作はKey- Preserved Table（実表のすべてのキーが、ビューでも同一のキーを保持する場合の実表）の列に対しては可能、削除操作はビューにKey- Preserved Tableが1つだけ存在する場合可能、挿入操作はKey- Preserved Tableの列に対して可能であり、更新・削除・挿入の各操作を実行する場合、各操作が反映される実表は1つであるという制限がある。

【0010】

40 【発明が解決しようとする課題】上記従来技術では、データベースのビュー定義を行うことによって実表の列名を仮想表した別名でのデータアクセスは可能である。

【0011】しかしながら、特に図3のように1つのデータベース（307）を複数アプリケーションから参照するデータベースアプリケーションシステムにおいて、アプリケーションプログラム（301、302、303）から直接データベースアクセスインターフェース（304、305、306）に命令を伝える形を取った場合、1つまたは複数のアプリケーションプログラム変更の際、他の変更されていないアプリケーションのコー

ド、またはデータベースのスキーマ定義に影響を与えてしまうこととなる。

【0012】アプリケーションのコードを変更した場合、再コンパイルの必要があり大きなコストがかかり、また、データベーステーブルの拡張や、レコードのカラムデータのアプリケーション内での意味付けの変更などの際、必要となるデータベース管理システムのビュー定義の変更を含むデータベーススキーマ定義の変更は、アプリケーションプログラムがデータベース管理者に依頼して行わなければならない、大きなコストがかかるという問題があった。

【0013】また、ビューを定義するSELECT文が「結合」の条件を含む場合、更新操作はKey-P reserved Tableの列に対してのみ可能、削除操作はビューにKey-P reserved Tableが1つだけ存在する場合にのみ可能、挿入操作はKey-P reserved Tableの列に対してのみ可能であり、更新・削除・挿入の各操作を実行する場合、各操作が反映される実表は1つであるという制限があり、データベースアプリケーション開発者はビューに対しての参照操作では実表の構造を意識せずにコーディングが可能であるが、更新・削除・挿入の各操作では実表の構造を意識したコーディングを行わなければならない、操作の一貫性に欠けるという問題があった。

【0014】本発明は、アプリケーション開発時に、レコードの更新・削除・挿入操作を行う場合もテーブルの構造を意識する必要を無くし、参照操作と、更新・削除・挿入操作のインターフェースに一貫性を持たせることを目的としており、さらにアプリケーション内で使用するレコードのカラムデータの変更や、カラムデータの意味付けの変更などの際、データベース管理システム内の定義情報の変更の必要を無くし、またテーブル構造の変更・拡張の際アプリケーションプログラムのコード変更・再コンパイルの必要を無くすことを目的とする。

【0015】

【課題を解決するための手段】上記課題を解決し、本発明の目的を達成するため、以下の手段が考えられる。1種類以上の実カラムデータを有する実レコードの集合である実テーブルを、複数備えているデータベースと、該データベースの実カラムデータを取得または変更するデータベース管理手段と、与えられたSQL文に対応して、前記データベース管理手段の該実カラムデータの取得または変更動作を制御するアクセス制御手段からなるデータベースアクセスシステムにおいて、該データベースの1つ以上の実テーブルのいくつかの実カラムデータを、1つの仮想表の各仮想表カラムデータとして関連付ける仮想表定義情報と、該仮想表カラムデータを取得または変更する仮想表操作命令を入力とし、該データベースの1つ以上の実テーブルの実カラムデータに対して操作を行う1つ以上のSQL文を生成し、該SQL文を、該アクセス制御手段に与えることにより、実カラムデー

タの取得または変更を操作し、該実カラムデータの取得または変更の結果を、仮想表カラムデータの取得または変更結果として出力する、データベース操作命令変換処理プログラムを持つことを特徴とする、データベースアクセスシステムを提供することにより、達成される。

【0016】すなわち、本発明は、図4のようにリレーショナルデータベースを用いたデータベースアプリケーションシステムにおいて、リレーショナルデータベースに格納されたデータを用いたデータベースアプリケーションプログラム(401, 402, 403)と、リレーショナルデータベース管理システムへのSQL文でのアクセスを提供するデータベースアクセスインターフェースプログラム(404, 405, 406)の間にリレーショナルデータベースの1つまたは複数のテーブルを、プログラム外部の仮想表定義ファイル(409, 411, 413)図5(501)に示す構成の502から取得する図6に示す仮想表定義情報テーブル(604)の構成の仮想表カラム名・テーブルカラム名対照情報(601)及びテーブルの結合カラム名集合情報(602)と、(503)から取得するテーブル内レコードユニーク性保証カラム名集合情報(603)から構成される仮想表定義情報テーブル(414, 415, 416)をもとに仮想表に変換し、アプリケーションプログラム(401, 402, 403)からの該仮想表に対する検索、挿入、変更、削除の操作命令の入力を、該仮想表へ該操作の結果を反映する、1つまたは複数のテーブルに対して操作を行う1つまたは複数のSQL文に変換し、生成されたSQL文をデータベースアクセスインターフェースプログラム(404, 405, 406)に対して発行することによって、テーブル構造をアプリケーションプログラム(401, 402, 403)に対して仮想表として仮想化する命令変換プログラム(408, 410, 412)を配置する。

【0017】命令変換プログラム(408, 410, 412)は仮想表レコード検索操作命令に対して、検索対象仮想表カラム名列と検索条件の入力から、仮想表定義情報テーブル(414, 415, 416)に格納されている情報と、入力情報中出现する仮想表カラム名から、検索対象とするテーブル名を解決し、入力の検索対象仮想表カラム名列と検索条件中出现する仮想表カラム名を、テーブルのカラム名に変換し、仮想表定義情報と検索対象テーブル名より、テーブル結合条件を生成し、テーブルのカラム名に変換した検索対象仮想表カラム名列、テーブルのカラム名に変換した検索条件、検索対象とするテーブル名、テーブル結合条件から、SQLのSELECT文を生成し、リレーショナルデータベースのデータベースアクセスインターフェースを用いて生成した検索命令を発行し、該検索結果のカラム名を、仮想表カラム名に変換して返す。

【0018】また、仮想表レコード挿入操作命令に対し

て、仮想表定義情報テーブルの情報と、仮想表カラム名とカラムデータの対の列からなる挿入レコード情報の入力から、該挿入レコード情報を各挿入対象テーブル単位に分割し、該テーブル単位挿入レコードの各仮想表カラム名をテーブルのカラム名に変換し、各挿入対照の候補となるテーブルに対し、整合性確認のSQLのSELECT文を生成して仮想表レコード挿入後の整合性の確認を行い、整合性が保たれている場合、各テーブル毎のSQLのINSERT文を生成し、該リレーショナルデータベースのデータベースアクセスインターフェースを用いて生成したSQLのINSERT文を発行する。

【0019】また、仮想表レコード削除操作命令に対して、仮想表定義情報と、削除条件の入力から、仮想表を構成するテーブルの全ての組に関して削除後のデータ整合性が保証できる削除条件を生成し、列挙済み削除条件をもとに、各テーブル毎のSQLのDELETE文を生成し、該リレーショナルデータベースのデータベースアクセスインターフェースを用いて生成したSQLのDELETE文を発行する。

【0020】また、仮想表レコード更新操作命令に対して、仮想表定義情報と、仮想表カラム名と仮想表カラムデータの組から成る仮想表更新レコード情報と、更新条件から、仮想表全カラム名を検索対象仮想表全カラム名、更新条件を検索条件入力とする仮想表レコード検索操作を行い、検索結果仮想表レコードの仮想表更新レコード情報に含まれる仮想表カラムのデータを仮想表更新レコード情報の仮想表カラムデータで置換し、該置換後検索結果仮想表レコードを入力として、前記仮想表レコード挿入操作を行う。

【0021】

【発明の実施の形態】以下、本発明の実施の形態を詳細に説明する。

【0022】図1は本発明を適用したデータベースアプリケーションシステムのブロック図であり、図2は、アプリケーションプログラムからの命令に対する処理全体の流れを表すフロー図であり、図6は仮想表定義ファイルの構成図であり、図7は対象とする操作対象とするデータベースのテーブル構成の1例であり、図8は図7のテーブル構成を対象として仮想表を構成する仮想表定義ファイルの1例である。

【0023】図1において、サーバマシン(115)はCPU(102)とキーボード(103)と外部記憶装置(104)と通信装置(105)と主記憶装置(107)及びバス(106)で構成され、外部記憶装置(104)にリレーショナルデータベース(RDB)(120)を格納し、主記憶装置(107)上にリレーショナルデータベース管理システム(RDBMS)(122)をロードし実行する。

【0024】またクライアントマシン(116)はCPU(109)とキーボード(111)と外部記憶装置

(110)と通信装置(108)と主記憶装置(114)及びバス(113)で構成され、外部記憶装置(110)上に仮想表定義ファイル(121)を格納し、主記憶装置(114)上にデータベースアプリケーションプログラム(AP)(117)と、命令変換プログラム(123)と、仮想表設定プログラム(119)をロードし実行する。命令変換プログラム(123)は、外部記憶装置(110)から仮想表定義ファイル(121)の情報を、仮想表定義情報テーブル(118)に読み込む。

【0025】仮想表定義情報は、図5で示す構成の仮想表定義ファイル(501)から、図6で示す構造を持つ仮想表定義情報テーブル(604)に格納する。仮想表定義ファイル(501)の「仮想表カラム名・テーブルカラム名対照情報、テーブル結合カラム名集合情報(502)」は仮想表定義情報テーブル(604)の仮想表カラム名・テーブルカラム名対照情報(601)に格納し、また1つの仮想表カラム名に複数のテーブルのカラム名が対応する場合は、テーブルの結合カラム名情報(602)にも格納する。また(501)のテーブル内レコードユニーク性保証カラム名集合情報(503)は仮想表定義情報テーブル(604)のテーブル内レコードユニーク性保証カラム名集合情報(603)に格納する。

【0026】図2において、命令変換プログラム(202)は、仮想表設定プログラム(204)を使用してあらかじめ作成された、仮想表定義ファイル(203)の情報を仮想表定義情報テーブル(208)に読み込む。AP(201)が命令変換プログラム(202)に仮想表レコードの検索操作または挿入操作または削除操作または更新操作の命令を発行すると、該仮想表レコード操作命令を該仮想表定義情報をもとに、1つ以上のSQLのSELECT文、またはINSERT文、またはDELETE文の組み合わせに変換し、対象とするリレーショナルデータベースのアクセスインターフェース(207)に対して発行し、RDBMS(205)を通してRDB(206)に操作を行う。

【0027】以下、図7に示すテーブル構成を例に、仮想表レコードの検索、挿入、削除、更新命令に対して、命令変換プログラム(202)が実行する処理を説明する。

【0028】図8(801)はテーブルA(701)、テーブルB(702)、テーブルC(703)から仮想表V(704)を構成するための仮想表定義ファイルの例であり、図9(704)は仮想表定義ファイル(801)を読み込んで生成する仮想表定義情報テーブルである。

【0029】図10は、命令変換プログラムの構成図である。以下、命令変換プログラムの仮想表レコード検索処理命令に対する動きを、仮想表V(704)を対象と

10

20

30

40

50

している命令変換プログラム(1020)に対し、図12に示す仮想表レコード検索命令(1200)を検索対象仮想表カラム名列(1201)及び仮想表レコード検索条件(1202)を入力として発行した場合を例にして説明する。

【0030】図11は検索命令に対する処理を表すフロー図である。(ステップ1100)において、仮想表定義情報読み込み部(1016)は仮想表定義ファイル(1017)を読み込み、仮想表定義情報テーブル(1015)に格納する。(ステップ1101)で、命令受付部(1100)は検索命令として、検索対象仮想表カラム名列(1201)及び、仮想表レコード検索条件(1202)の入力を受ける。

【0031】(ステップ1102)では、(ステップ1101)で受け取った検索対象仮想表カラム名列(1201)を検索対象仮想表カラム名処理部(1002)に、仮想表レコード検索条件(1202)を検索条件処理部(1003)にそれぞれ渡す。ここで検索対象仮想表カラム名処理部(1002)、及び検索条件処理部(1003)は、それぞれの入力中に現れる仮想表カラム名を、検索対象カラム名管理部(1012)に登録する。ここで(1200)の入力で登録される仮想表カラム名は(1203)の通りである、全出現仮想表カラム名を登録し終わり次第、SQL生成部(1001)に制御が移る。

【0032】(ステップ1103)で、SQL生成部(1001)は検索対象テーブル名・カラム名決定部(1004)に検索対象テーブル名の解決を依頼する。(1200)の入力で(ステップ1103)では(1204)から検索対象となるテーブル(1204)と決定する。

【0033】(ステップ1104)で、SQL生成部は検索対象仮想表カラム名処理部(1002)と仮想表レコード検索条件処理部(1003)を呼び出し、カラム名変換後の検索対象カラム名列と検索条件を取得する。この時、検索対象仮想表カラム名処理部(1002)及び仮想表レコード検索条件処理部(1003)は検索対象テーブル名・カラム名決定部(1004)を呼び出し、検索対象とするテーブル名を取得し、カラム名の変換を行う。ここで検索対象カラム名については、それぞれの検索対象カラム名に続けてダブルクォーテーションで囲むことによって、検索結果のカラム名が仮想表カラム名となるようにする。(1200)の入力に対しては、(1206)のそれぞれ検索対象カラム名(1207)、検索条件(1208)の出力を得る。

【0034】(ステップ1105)ではSQL生成部がテーブル結合条件生成部(1005)を呼び出し、検索対象となるテーブルについての結合条件を取得する。テーブル結合条件生成部(1005)では検索対象テーブル名・カラム名決定部(1004)より検索対象テブ

ル名を取得し、仮想表定義情報テーブル(1015)のテーブル結合カラム名情報をもとに、テーブル結合条件を生成する。(1200)の入力に対しては、(1209)が生成されることとなる。

【0035】(ステップ1106)では、(ステップ1103)で生成された検索対象テーブル名、(ステップ1104)で生成された検索対象カラム名、検索条件、(ステップ1105)で生成されたテーブル結合条件をもとにSQLのSELECT文を生成する。(1200)の入力に対しては(1210)が生成される。

【0036】(ステップ1107)では、SQL命令発行部(1010)においてSQL命令生成部(1001)で生成されたSQL文をデータベースアクセスインターフェース(1018)に対し発行する。(ステップ1108)では、検索結果処理部(1011)において、検索結果をアプリケーションで使用する形式に変換し返す。

【0037】次に、命令変換プログラムの仮想表レコード挿入処理命令に対する動きを、仮想表V(704)を対象としている命令変換プログラム(1020)に対し、図14に示す仮想表レコード挿入命令(1400)を仮想表カラム名(1401)及び仮想表カラムデータ(1402)を入力として発行した場合を例にして説明する。

【0038】図13は挿入命令に対する処理を表すフロー図である。(ステップ1300)において、仮想表定義情報読み込み部(1016)は仮想表定義ファイル(1017)を読み込み、仮想表定義情報テーブル(1015)に格納する。(ステップ1301)で、命令受付部(1100)は挿入命令として、仮想表カラム名(1401)と、仮想表カラムデータ(1402)からなる仮想表レコード挿入命令(1400)の入力を受ける。

【0039】(ステップ1302)では仮想表レコード分割処理部(1006)において、各カラムに対応しているテーブル毎に仮想表レコードの分割を行い、挿入レコード情報管理テーブル(1013)に登録する。

【0040】ここで仮想表レコード挿入命令(1400)が入力であった場合は、(1406)のようになり、それぞれテーブルAに対応する部分(1403)、テーブルBに対応する部分(1404)、テーブルCに対応する部分(1405)である。ここでテーブルA、テーブルB、テーブルCに関して、未処理と設定する。

【0041】(ステップ1303)では、トランザクションを開始する。これは仮想表レコード挿入操作はSQLレベルでは1つ以上のデータ変更命令で構成されるのに対して、命令変換プログラムの外部に対しては1つの操作として提供しているため、命令変換プログラム中での全てのデータベース変更処理を合わせて1つのアトミックな操作とする必要があるからである。

10

20

30

40

50

【0042】(ステップ1304)では、挿入整合性確認部(1007)において挿入レコード情報管理テーブル(1013)を参照し、データ挿入対象でありまだ処理されていないテーブルがあるかを調べる。図14の例ではテーブルA、テーブルB、テーブルCが未処理であるので、YESである。

【0043】(ステップ1305)では挿入対象テーブルであり、未処理のテーブルを1つ選択しTとする。図14の例では任意でテーブルAをTとする。(ステップ1306)では(ステップ1305)で選択したテーブルにテーブル内レコードユニーク性保証カラムがあるかを仮想表定義情報テーブルを参照して検査する。ここで例のテーブルAではIDカラムがユニーク性保証カラムとなっているのでYESとなる。

【0044】(ステップ1309)ではTのユニーク性保証カラムがTに関しての挿入レコード情報に全て存在しているかを検索する。例ではAのユニーク性保証カラムはIDのみであり、Aに関しての挿入レコード情報(1403)内に全て存在していることになり、YESとなる。存在していない場合は、挿入した場合の整合性保証が行えなくなるので、トランザクションをロールバックし(ステップ1314)、終了する。

【0045】(ステップ1310)ではTに関しての挿入レコード情報のユニーク性保証カラムのデータと等価のレコードが存在するか否かを検索する。この時、検査に使用するSQLは、挿入レコード情報の[カラム名=カラムデータ]の条件式をカラム数だけAND条件でつないだ条件式を持ったSELECT文であり、検索対象カラムは全カラムとする。図14での例の場合、検査用のSQLは(1407)となる。

【0046】ここで検索結果は0件であった場合は、存在していないレコードとなるので、レコードを挿入(ステップ1307)してもよいことになる。(ステップ1312)では(ステップ1311)において検索した結果のレコードと、挿入レコード情報を比較する。異なる場合は不整合が発生することを意味するので、トランザクションをロールバックし(ステップ1314)、終了する。同値であった場合は、既に同一のレコードが存在することを意味し、レコードの挿入を行う必要はない。

【0047】例では検索した結果のレコードと、挿入レコード情報は同値であるので、テーブルAにレコードの挿入を行う必要はないことがわかる。(ステップ1308)では、Tを処理済に設定する。以下同様にテーブルB、テーブルCについて行う。その結果、テーブルBに関しては1408、テーブルCに関しては1409、1410のSQL文が生成される。テーブルB、テーブルCの処理が終了すると、(ステップ1304)の分岐において、NOとなり、トランザクションをコミットし(ステップ1313)、終了する。

【0048】次に、命令変換プログラムの仮想表レコー

ド削除処理命令に対する動きを、仮想表V(704)を対象としている命令変換プログラム(1020)に対し、図17に示す仮想表レコード削除命令(1700)を仮想表レコード削除条件入力として発行した場合を例にして説明する。

【0049】図15は削除命令に対する処理を表すフロー図である。(ステップ1500)において、仮想表定義情報読み込み部(1016)は仮想表定義ファイル(1017)を読み込み、仮想表定義情報テーブル(1015)に格納する。(ステップ1522)で、命令受付部(1100)は仮想表レコード削除命令として、仮想表レコード削除条件の入力を受ける。図17の例では(1700)を仮想表レコード削除条件入力とする。

【0050】(ステップ1501)では仮想表レコード条件処理部(1008)において入力の仮想表レコード削除条件に関連するテーブル毎に分割し、削除条件情報管理テーブル(1014)に格納する。ここで削除条件管理テーブル(1014)は、図16に示すように仮想表に含まれるテーブル数をNとしたとき $N \times N$ 構造を持っている削除条件テーブル(1600)と $1 \times N$ 構造を持つルートフラグテーブル(1601)からなる。

【0051】(ステップ1501)において、入力の仮想表レコード削除条件をテーブル毎に分割した条件はそれぞれ対応するテーブルをTとした場合、削除条件テーブル(1600)の(T, T)の位置に格納する。ルートフラグテーブル(1601)をOFFに初期化する。(1701)は(1700)の入力に対して(ステップ1501)終了時点での削除条件テーブル(1600)の内容を示したものであり、(1704)はルートフラグテーブルの内容を示したものである。

【0052】(ステップ1521)では、トランザクションを開始する。これは仮想表レコード削除操作はSQLレベルでは1つ以上のデータ変更命令で構成されるのに対して、命令変換プログラムの外部に対しては1つの操作として提供しているため、命令変換プログラム中での全てのデータベース変更処理を合わせて1つのアトミックな操作とする必要があるからである。

【0053】(ステップ1502)では、ルートフラグテーブル(1601)の対応する要素がOFFとなっているテーブルが存在するかどうかを検査する。存在しない場合は(ステップ1509)へ進む。(ステップ1503)ではルートフラグがOFFのテーブルの中から任意の1つを選び、ルートとし、対応するルートフラグテーブル(1601)の要素をONとする。また、カレントフラグテーブルをOFFに初期化する。

【0054】(ステップ1504)では削除条件テーブル(1600)のルートテーブルに対応する行の対応するカラムに削除条件が入っており、カレントフラグがOFFのテーブルが存在するかを検査する。存在した場合はその中から1つ選びカレントし、カレントフラグをO

10

20

30

40

50

Nにし(ステップ1505)、カレントの結合対象であり、かつ削除条件テーブル(1600)のルートに対応する行の、対応するカラムに条件が入っていないテーブルが存在するかを検査する(ステップ1506)。

【0055】存在しない場合は(ステップ1504)へ。存在する場合は、該当する任意のテーブルをジョインに設定し、削除条件テーブル(1600)の(カレント、ルート)を条件として、カレントとジョインの結合カラムを検索対象列とする検索を実行し(ステップ1508)、検索結果が存在するかを検査する(ステップ1519)。存在しなかった場合は、トランザクションをロールバックし(ステップ1520)、終了する。

【0056】存在した場合は、検索結果レコードについて、それぞれ重複数を数え(ステップ1518)、検索結果レコードをもとに「カラム名=カラムデータ」形式の条件をAND結合でレコードカラム数だけ結合したものを検索条件とし、検索対象カラムを全カラムとする検索を各レコードに関して行い、そのヒット数を数える(ステップ1517)。

【0057】ここで(ステップ1518)で算出された重複数と(ステップ1517)で求められたヒット数が一致するレコードが存在するかを検査する(ステップ1515)。存在しない場合は削除条件テーブル(1600)のジョイン、ルート)に削除不能フラグを設定し(ステップ1516)、(ステップ1506)へ進む。存在する場合は、一致するレコードそれぞれについて「カラム名=カラムデータ」形式の条件をAND結合でレコードカラム数だけ結合した条件を生成し、各生成された条件をOR結合で結合した条件を削除条件テーブル(1600)の(ジョイン、ルート)に格納し(ステップ1514)、(ステップ1506)へ進む。

【0058】ここで(1702)は(1700)の入力に対して構成される削除条件テーブルの内容である。また(1705)は(1702)が構成された時点での、ルートフラグテーブルの内容である。(ステップ1502)において、ルートフラグテーブルの値がOFFのテーブルが存在しなかった場合、削除条件テーブル(1600)の各列について、各行の条件式をAND結合で結合し対応するテーブルに対し発行する(1513)。

【0059】ここで削除条件テーブルの対応する列に条件が1つも入っていないもしくは削除不能フラグが設定されているテーブルに関しては、操作を行わない(ステップ1512)。ここで(1703)は(1700)の入力に対して生成される、各テーブルに対する削除の条件である。全テーブルの処理が終わったところでトランザクションをコミットし(ステップ1510)、終了する。

【0060】次に、命令変換プログラムの仮想表レコード更新処理命令に対する動きを説明する。

【0061】図18は更新命令に対する処理を表すフロ

ー図であり、図19は仮想表レコード更新命令の例である。仮想表定義情報読み込み部(1016)は仮想表定義ファイル(1017)を読み込み、仮想表定義情報テーブル(1015)に格納する(ステップ1800)。命令受付部(1100)は更新命令として、仮想表カラム名(1901)と、仮想表カラムデータ(1902)と、仮想表レコード更新条件(1903)からなる仮想表レコード更新命令(1900)の入力を受ける(ステップ1801)。

【0062】(ステップ1810)では、トランザクションを開始する。これは仮想表レコード更新操作はSQLレベルでは1つ以上のデータ変更命令で構成されるのに対して、命令変換プログラムの外部に対しては1つの操作として提供しているため、命令変換プログラム中での全てのデータベース変更処理を合わせて1つのアトミックな操作とする必要があるからである。

【0063】(ステップ1802)では仮想表レコード検索条件入力として仮想表レコード更新条件、検索対象仮想表カラム名として仮想表全カラム名を指定して、仮想表レコード検索処理を実行する。検索結果仮想表レコードが1件以上あることを検査し(ステップ1803)、0件の場合は更新対象レコードなしとして、トランザクションをロールバックし(ステップ1808)、終了する。

【0064】検索結果が存在する場合は、仮想表レコード削除条件入力として、仮想表レコード削除を実行する(ステップ1804)。(ステップ1805)では(ステップ1802)での検索結果集合が空であるか否かを検査する。空であった場合にはトランザクションをコミットし(ステップ1809)、終了する。

【0065】空でなかった場合は、任意の1つを取り出し、仮想表更新レコード情報に含まれるカラム名に対応するカラムデータを仮想表更新レコード情報のカラムデータに変換し(ステップ1806)、変換した仮想表レコードを挿入する仮想表レコードとして仮想表レコード挿入操作を実行し(ステップ1807)、(ステップ1805)へ進む。

【0066】上記実施例によれば、アプリケーションの対象テーブルをアプリケーションの実行者が、データベース管理システムの定義を変更することなく、変更することが可能となるという効果がある。

【0067】上記実施例において、命令変換プログラムが対象とする仮想表の指定方法は、以下のような方法が考えられる。

【0068】図20は、実行時の指定でアプリケーションが対象とするデータを変更する、命令変換プログラムを使用するアプリケーションの構成例である。外部記憶装置(2003)は命令変換プログラムを使用するアプリケーションの実行ファイル(2004)と、商品管理仮想表定義ファイル(2005)と、顧客管理仮想表定

義ファイル(2006)を格納する。

【0069】商品データベースアプリケーション実行手段(2001)を実行した場合、(2004)が主記憶装置(2007)にロードされ、商品管理仮想表定義ファイル(2005)の情報を商品管理仮想表定義情報テーブル(2010)に格納し、商品データ管理データベース(2012)を操作する商品データベースアプリケーション(2008)となる。

【0070】顧客データベースアプリケーション実行手段(2002)を実行した場合、(2004)が主記憶装置(2007)にロードされ、顧客管理仮想表定義ファイル(2006)の情報を顧客管理仮想表定義情報テーブル(2011)に格納し、顧客データ管理データベース(2013)を操作する顧客データベースアプリケーション(2009)となる。

【0071】上記実施例において、アプリケーションプログラムが対象とする仮想表を、立ち上げ時に定義ファイル名称を指定するという簡単な手段により、切り替えることができ、例えば上記実施例の商品データベースアプリケーションと顧客データベースアプリケーションの

ように、ロジックの同じアプリケーションプログラムを別のアプリケーションとして使用することが可能となる。

【0072】また、上記実施例における仮想表操作命令は仮想表に対するSQLであってもよい、この場合SELECT文及びDELETE文のFROM句、及び、INSERT文のINTO句内は、仮想表名となる。

【0073】本実施例においては、SQL構文が拡張された場合でも仮想表による効果が実現できる。

【0074】例えば、以下のような拡張の例が考えられる。

【0075】SELECTニュース番号、F1(ニュース、'記事、本文'、'記事[本文{"日立"}]')、価格
FROMニューステーブル
WHERE F2(ニュース、'記事[本文{"日立"}]') IS TRUE

ここで、F1、F2は拡張文法である。
【0076】この場合、先の実施例における、検索対象仮想表カラム名は、NewsNo、F1(News、'記事、本文'、'記事[本文{"日立"}]') VALUEに対応し、仮想表レコード検索条件はF2(News、'記事[本文{"日立"}]') IS TRUEに対応し、検索対象の実テーブルに対応するFROM句と、結合条件が生成される。

【0077】出力結果は以下ようになる
SELECT NewsMaster.NewsNo、F1(NewsMaster.News、'記事、本文'、'記事[本文{"日立"}]')、NewsPrice.Price
FROM NewsMaster、NewsPrice

WHERE F2(NewsMaster.News、'記事[本文{"日立"}]') IS TRUE AND NewsMaster.NewsNo=NewsPrice.NewsNo

【0078】

【発明の効果】以上述べたように、本発明によれば、データベースアプリケーション開発者は、データベース実体の複雑な構造を意識することなく、コンフィグレーションファイル定義によって仮想化された1つのテーブルに対しての操作のみを意識したコーディングを行うことができるので、データベースアプリケーション開発工数削減の効果がある。

【0079】また、プログラムコード中における操作対象カラム名には、プログラム外部のデータベースアクセス環境設定ファイル中において定義の、仮想化された名称が使用されるため、データベース実体の定義や、レコードのカラムデータのアプリケーション内での意味付けを変更した場合にも、プログラムのコードの変更や再コンパイルの必要はなく、データベースアクセス環境設定ファイルの内容変更のみで対応することができるので、保守コスト削減の効果がある。

【図面の簡単な説明】

【図1】本発明を適用したデータベースアプリケーションシステムのブロック図である。

【図2】アプリケーションプログラムからの命令に対する処理全体の流れを表すフロー図である。

【図3】1つのデータベースを複数アプリケーションから参照するデータベースアプリケーションシステムのブロック図である。

【図4】1つのデータベースを複数アプリケーションから参照する、本発明を適用したデータベースアプリケーションシステムのブロック図である。

【図5】仮想表定義ファイルの構成図である。

【図6】仮想表定義情報テーブルの構成図である。

【図7】実施の形態で操作対象として使用するデータベースのテーブル構成図である。

【図8】図7のテーブル構成を対象として仮想表を構成する仮想表定義ファイルの1例を示す図である。

【図9】図8の仮想表定義ファイルを読み込んで生成する仮想表定義情報テーブルを示す図である。

【図10】命令変換プログラムの構成を示すブロック図である。

【図11】検索命令に対する処理を表すフロー図である。

【図12】検索命令に関するデータの例を示すフロー図である。

【図13】挿入命令に対する処理を表すフロー図である。

【図14】挿入命令に関するデータの例を示すフロー図である。

【図15】削除命令に対する処理を表すフロー図であ

る。

【図16】削除条件管理テーブルの構成を表すブロック図である。

【図17】仮想表レコード削除命令の例である。

【図18】更新命令に対する処理を表すフロー図である。

【図19】仮想表レコード更新命令の例を示すブロック図である。

【図20】実行時の指定でアプリケーションが対象とするデータを変更する命令変換プログラムを使用するアプリケーションの構成例を示す図である。

【符号の説明】

201…アプリケーションプログラム、202…命令変換プログラム、203…仮想表定義ファイル、

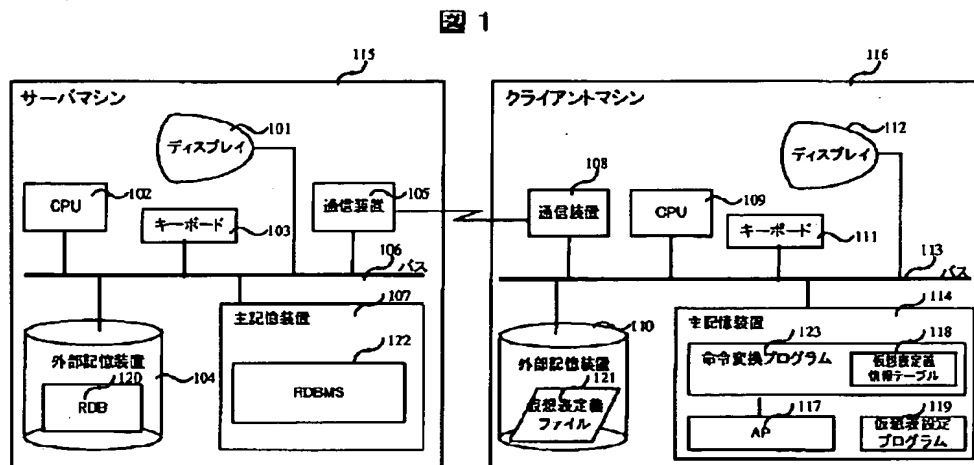
204…仮想表設定プログラム、205…データベース管理装置、206…データベース、207*

*…アクセスインターフェース、208…仮想表定義情報テーブル、1000…命令受付部、

1001…SQL生成部、1002…検索対象仮想表カラム名処理部、1003…仮想表レコード検索条件処理部、1004…検索対象テーブル名・カラム名決定部、1005…テーブル結合条件生成部、1006…仮想表レコード分割処理部、1007…挿入整合性確認部、1008…仮想表レコード条件分割処理部、1009…削除整合性保証部、1010…SQL命令発行部、1011…検索結果処理部、

1012…検索対象仮想表、1013…カラム名管理部、1014…挿入レコード情報管理テーブル、1015…仮想表定義情報テーブル、1016…仮想表定義情報読み込み部、1017…仮想表定義ファイル、1018…データベースアクセスインターフェース、1019…命令変換プログラム。

【図1】

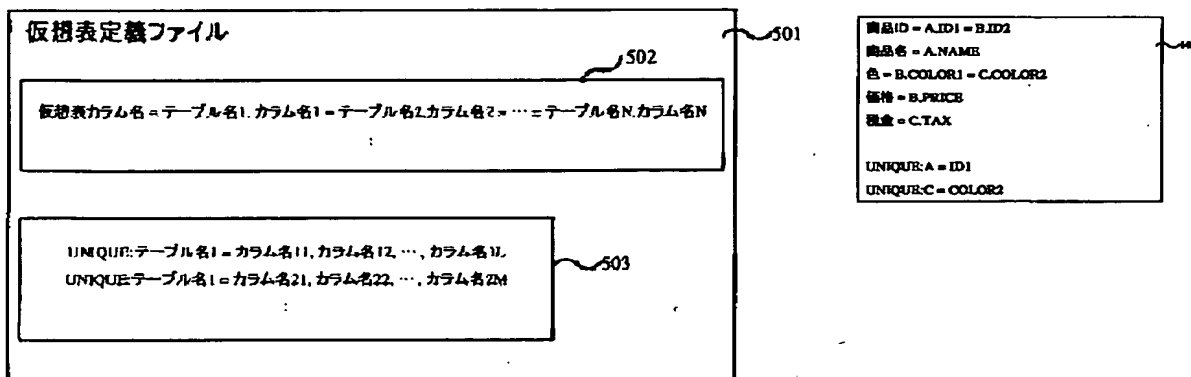


【図5】

【図8】

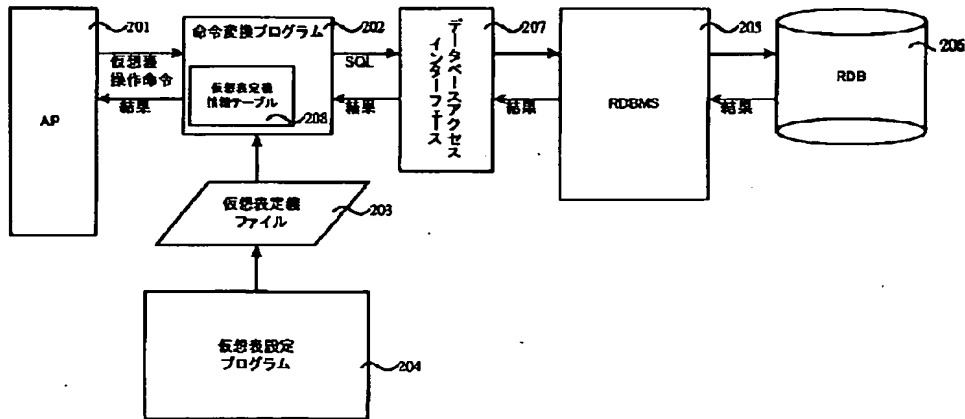
図5

図8



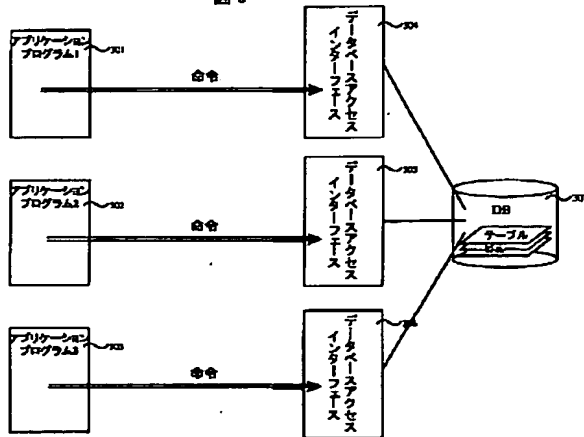
【図2】

図2



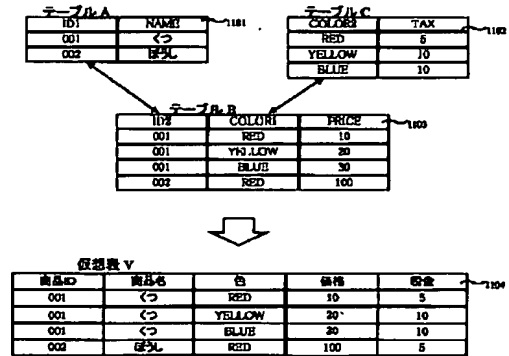
【図3】

図3



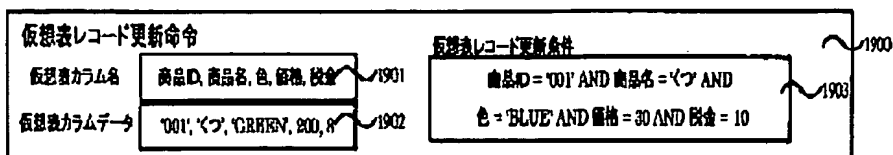
【図7】

図7

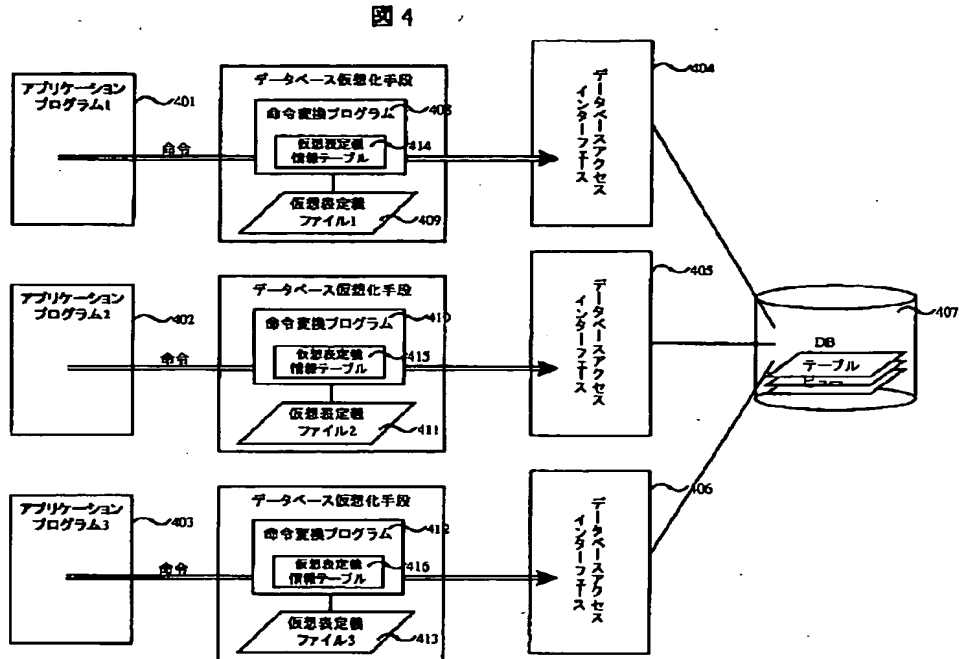


【図19】

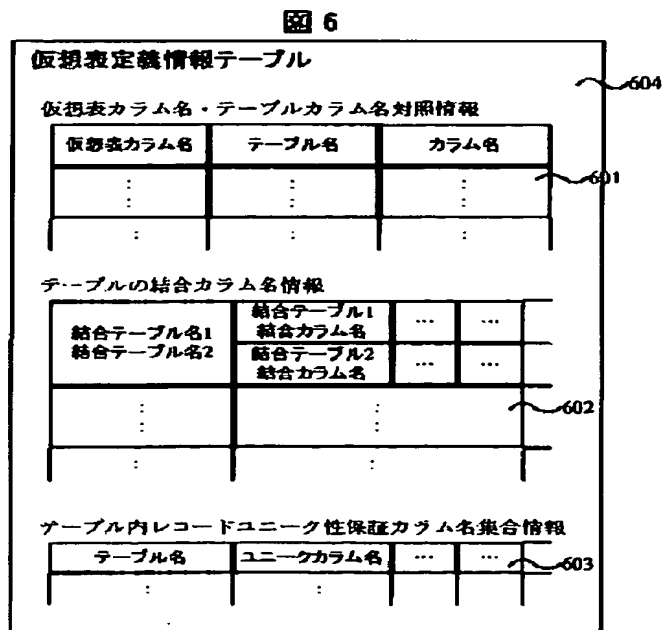
図19



【図4】

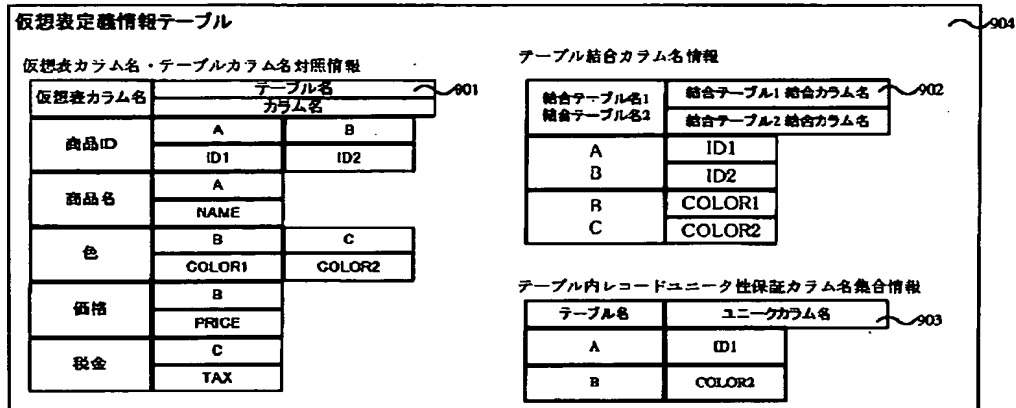


【図6】



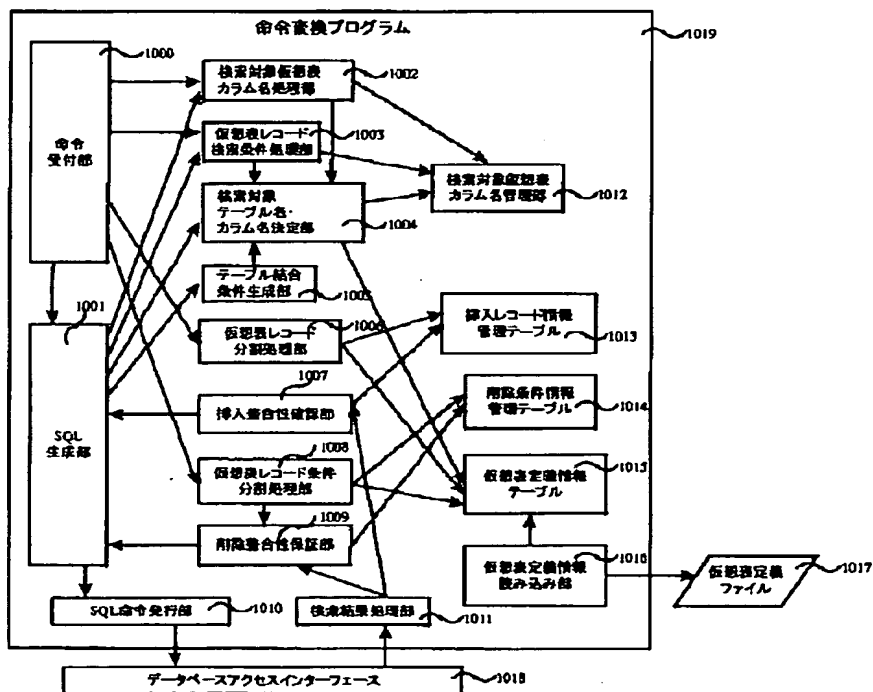
【図9】

図9



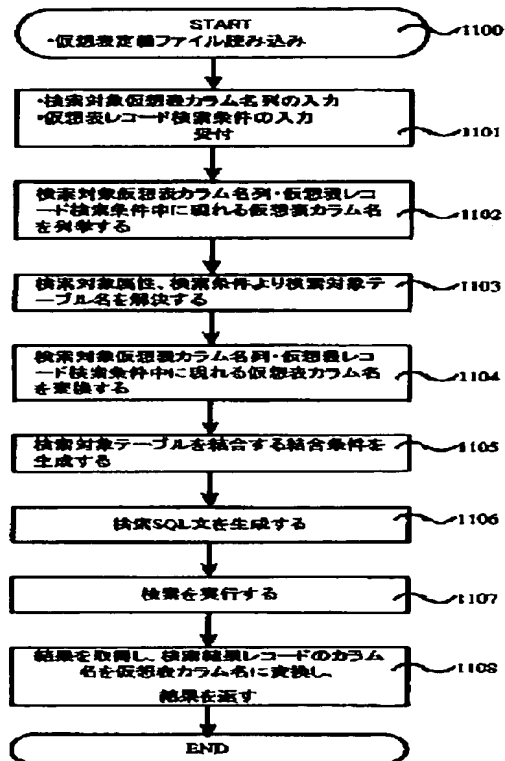
【図10】

図10



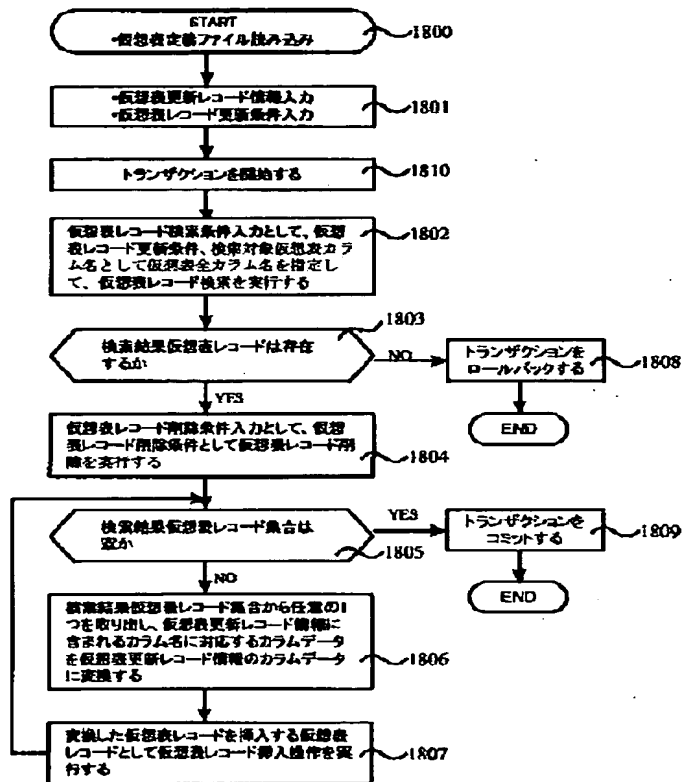
【図11】

図 11



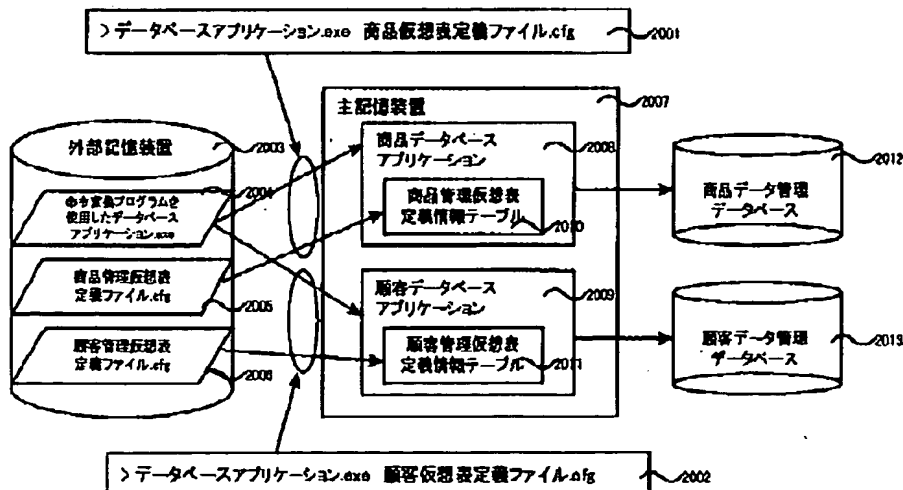
【図18】

図 18



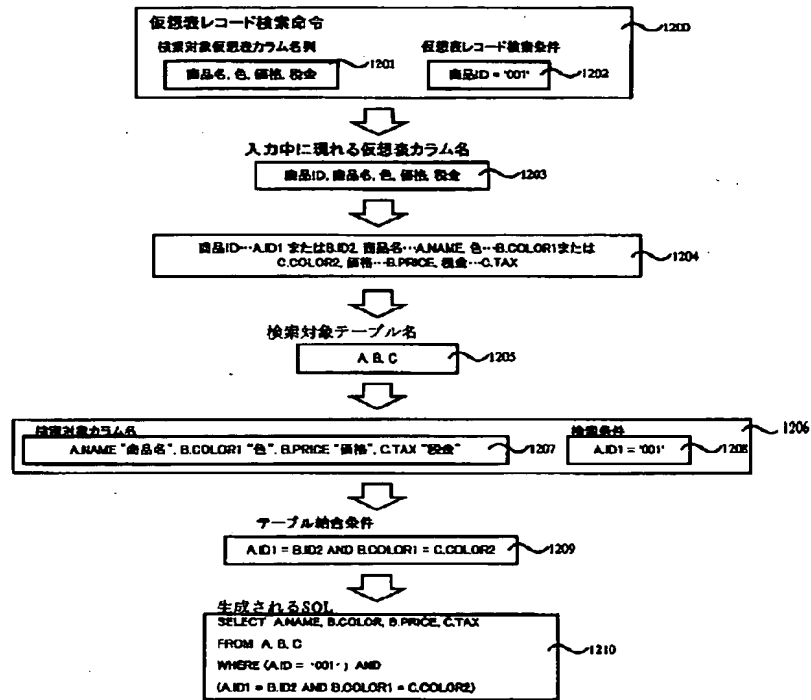
【図20】

図 20



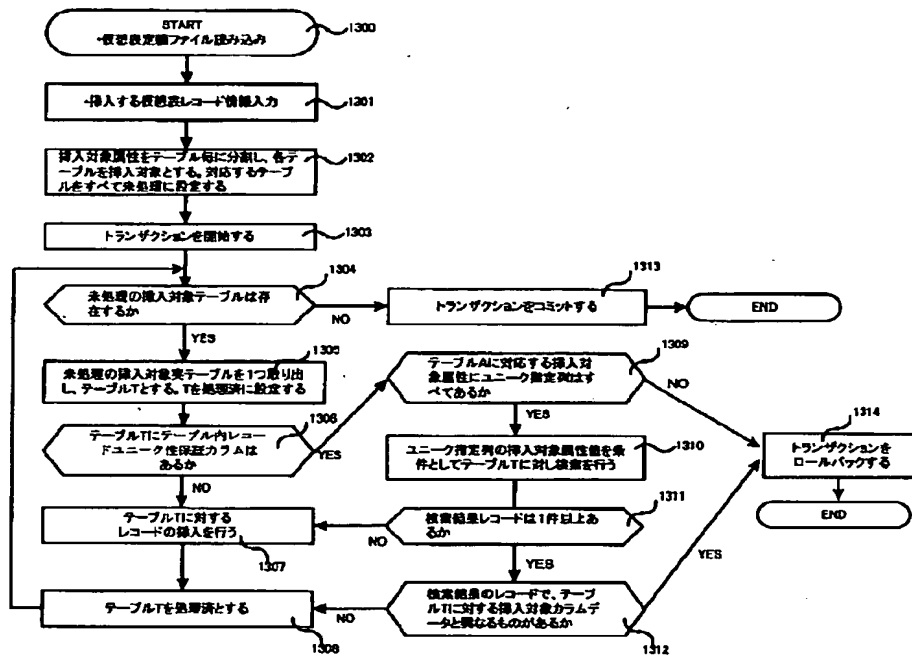
【図12】

図 12



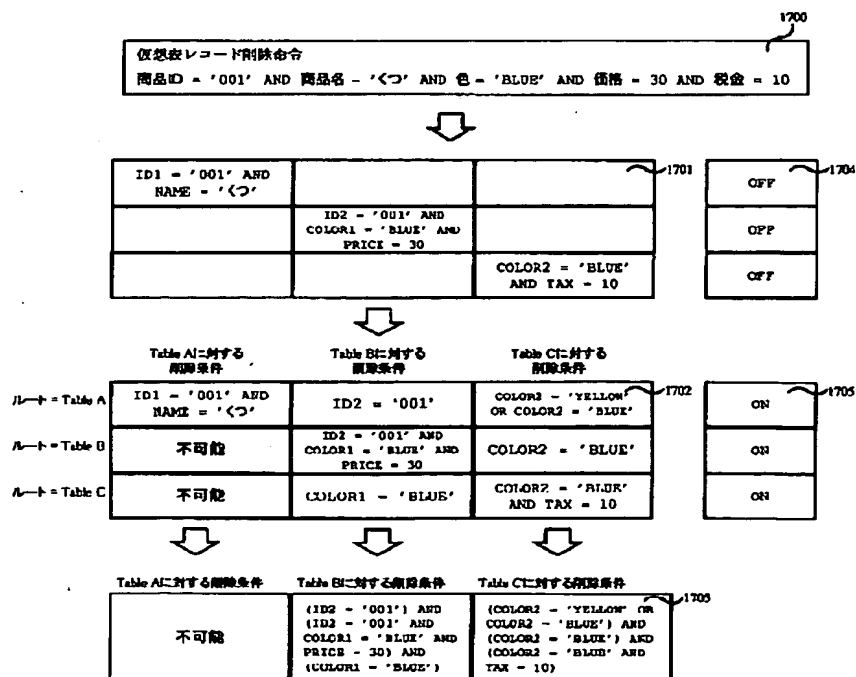
【図13】

図 13



【図17】

図 17



【図14】

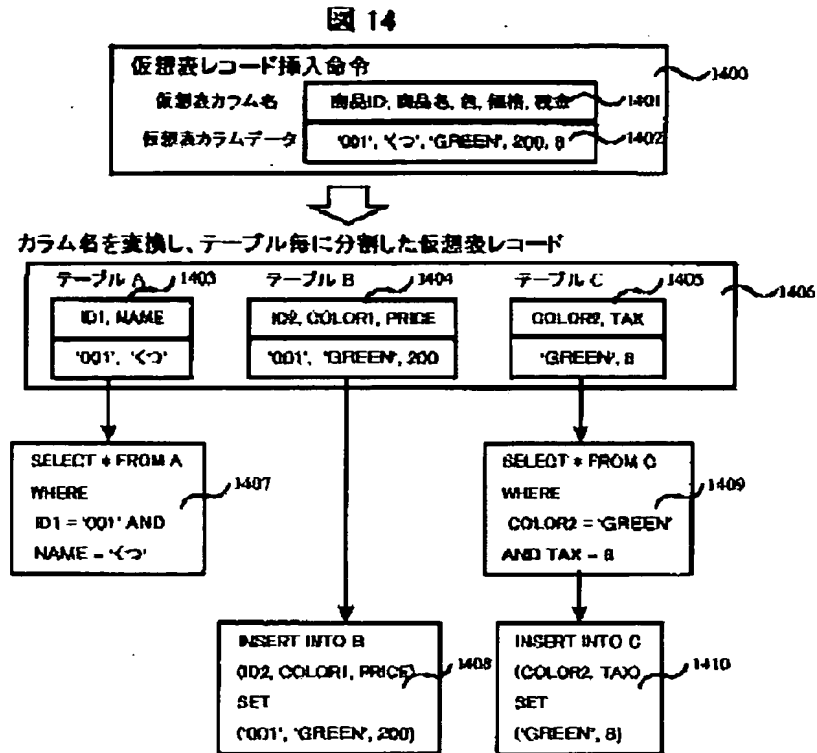
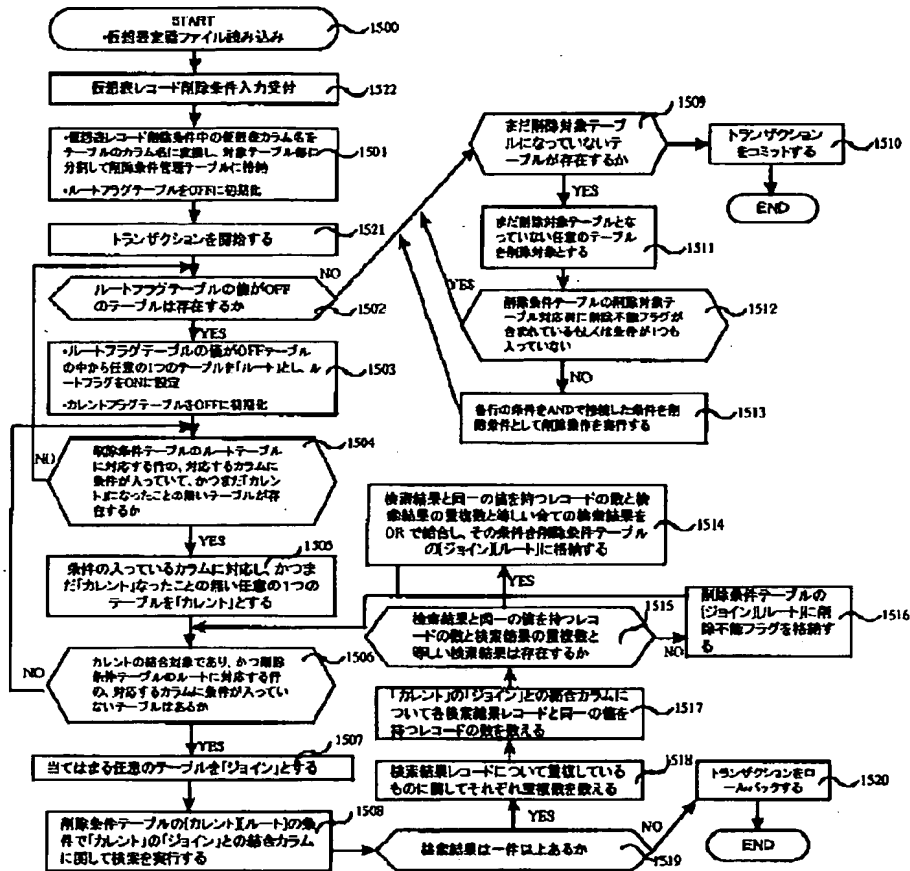
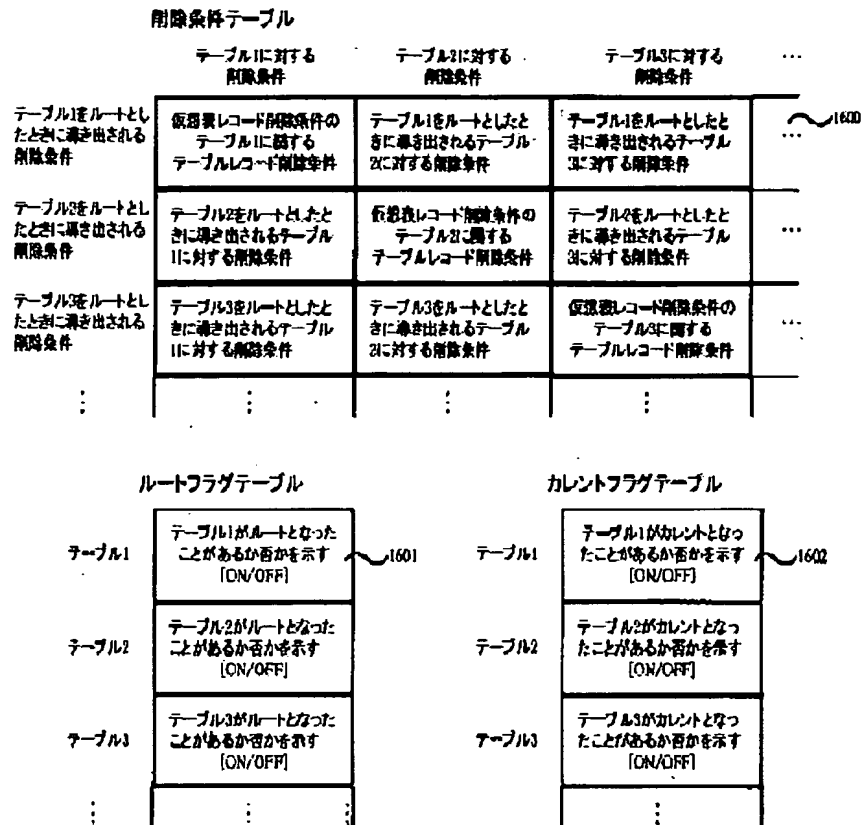


图 15



【図16】

図 16



フロントページの続き

(72)発明者 三堀 潔

神奈川県戸塚区戸塚町5030番地 株式会社

日立製作所ソフトウェア開発本部内

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 11-265395

(43)Date of publication of application : 28.09.1999

(51)Int.Cl.

G06F 17/30

G06F 12/00

(21)Application number : 10-068099

(71)Applicant : HITACHI LTD

(22)Date of filing : 18.03.1998

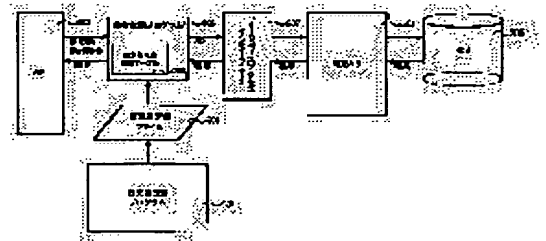
(72)Inventor : SUZUKI TOYOHITO
KOIKE HIROSHI
MIZOTE YUJI
MITSUBORI KIYOSHI

(54) DATA BASE ACCESS SYSTEM

(57)Abstract:

PROBLEM TO BE SOLVED: To enable an application executant to change an objective table for application without changing the definition of a data base management system.

SOLUTION: The data base access system is provided with a data base operation instruction converting processing program 202 for generating one or more structured query language(SQL) sentences for operating real column data in one or more real tables stored in a data base, acquiring or changing the real column data by applying these SQL sentences to an access control means and outputting the acquired or changed result of the real column data as the acquired or changed result of virtual table column data.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

Copyright (C); 1998,2003 Japan Patent Office

* NOTICES *

Japan Patent Office is not responsible for any damages caused by the use of this translation.

- 1.This document has been translated by computer. So the translation may not reflect the original precisely.
- 2.**** shows the word which can not be translated.
- 3.In the drawings, any words are not translated.

CLAIMS

[Claim(s)]

[Claim 1] A data base equipped with two or more real tables which are the sets of a real record which have one or more kinds of real column data A database management means to acquire or change real column data of this data base, and an access-control means to control acquisition or modification actuation of this real column data of said database management means corresponding to a given SQL sentence Virtual table definition information which is the data base access system equipped with the above, and associates some real column data of one or more real tables of this data base as each virtual table column data of one virtual table, Virtual table operating instructions which acquire or change this virtual table column data are considered as an input. By generating one or more SQL sentences which operate it to real column data of one or more real tables of this data base, and giving this SQL sentence to this access-control means It is characterized by having the data-base-manipulation instruction transform-processing program which operates acquisition or modification of real column data, and outputs a result of acquisition of this real column data, or modification as acquisition or a modification result of virtual table column data.

[Claim 2] A data base equipped with two or more real tables which are the sets of a real record which have one or more kinds of real column data, A database management means to acquire or change real column data of this data base, It is aimed at an access-control means to control acquisition or modification actuation of this real column data of said database management means, corresponding to a given SQL sentence. Virtual table definition information which supposes some real column data of one or more real tables of this data base as each virtual table column data of one virtual table, Virtual table operating instructions which acquire or change this virtual table column data are considered as an input. By generating one or more SQL sentences which operate it to real column data of one or more real tables of this data base, and giving this SQL sentence to this access-control means A data-base-manipulation instruction transform-processing program which operates acquisition or modification of real column data, and outputs a result of acquisition of this real column data, or modification as acquisition or a modification result of virtual table column data.

[Claim 3] They are a data-base-manipulation instruction transform-processing program characterized by constituting from contrast information with which said virtual table definition information relates a virtual table column name and a real table column name in a data-base-manipulation instruction transform-processing program according to claim 1 or 2, and set information on a real column name which associates one or more real tables which constitute this virtual table, and a data base access system.

[Claim 4] In claim 1 or two publications, it becomes an input to said data-base-manipulation instruction transform-processing program. Contrast information which associates some real column data of one or more real tables of a data base as each virtual table column data of one virtual table, and associates a virtual table column name and a real table column name for constituting a virtual table, Data medium characterized by recording virtual table definition information which consists of a set of a real column name which associates one or more real tables which constitute this virtual table.

[Claim 5] A data base access system by which this data-base-manipulation instruction transform-processing program is characterized by determining the target virtual table by conferring upon a starting instruction of said data-base-manipulation instruction transform-processing program a discernment name of data medium which recorded virtual table definition information, and starting it in claim 1 or two publications.

[Claim 6] In claim 1 or two publications, virtual table record retrieval operating instructions to said virtual

table An input of virtual table KOREDO retrieval conditions to a virtual table column name train and this virtual table definition information used as a candidate for retrieval, From a virtual table column name which appears in this input, a real table name made applicable to retrieval is solved. Change into a real table column name a virtual table column name which appears in this virtual table column name train for retrieval, and it considers as a real table column name train for retrieval. A virtual table column name which appears in this virtual table record retrieval condition is changed into a real table column name, and it considers as real table record retrieval conditions. This virtual table definition information, A table connection condition is generated from a real table name made applicable [this] to retrieval. This real table column name train for retrieval, These real table record retrieval conditions and a real table name made applicable [this] to retrieval, A SELECT sentence of SQL which generated a SELECT sentence of SQL and was this generated from this real table connection condition using an access-control means of this data base is published. A data base access system characterized by having the data-base-manipulation instruction conversion program which searches a virtual table record by changing and returning a column name of this retrieval result to a virtual table column name.

[Claim 7] In claim 1 or two publications, virtual table record insertion operating instructions to said virtual table From an input of said virtual table definition information and insertion virtual table record information which consists of a virtual table column name and a pair of train of virtual table column data Divide this insertion virtual table record information into each real table unit for insertion, and each virtual table column name of this real table unit inserted record is changed into a real table column name. To a real table which serves as a candidate of each insertion contrast, generate a SELECT sentence of SQL of an adjustment check and adjustment after virtual table record insertion is checked. By publishing an INSERT sentence of SQL which generated an INSERT sentence of SQL for every fruit table, and was this generated using an access-control means of this data base, when adjustment is maintained A data base access system characterized by having the data-base-manipulation instruction conversion program which inserts this virtual table record.

[Claim 8] In claim 1 or two publications, virtual table record deletion operating instructions to said virtual table A virtual table column name in a virtual table record deletion condition is changed into a real table column name from said virtual table definition information and an input of virtual table record deletion conditions. Deletion conditions of a real table record that data consistency after deletion can be guaranteed about all groups of a real table which is applicable and which divides for every real table and constitutes this virtual table are generated. By publishing a DELETE sentence of SQL which generated a DELETE sentence of SQL for every table, and was this generated using an access-control means of this data base based on these real table record deletion conditions A data base access system characterized by having the data-base-manipulation instruction conversion program which deletes a virtual table record.

[Claim 9] In claim 1 or two publications, a virtual table record update operation instruction to said virtual table Said virtual table definition information and virtual table update record information which consists of a group of a virtual table column name and virtual table column data, From renewal conditions of a virtual table record, all virtual table column names A virtual table column name for retrieval, Said virtual table record retrieval actuation which considers these renewal conditions of a virtual table record as a virtual table record retrieval condition input is performed. Virtual table column data of this virtual table update record information replaces data of a virtual table column contained in this virtual table update record information on this retrieval result virtual table record. By performing said virtual table record deletion actuation which considers these renewal conditions of a virtual table record as a virtual table record deletion condition input, and performing said virtual table record insertion actuation by considering this after [substitute] retrieval result virtual table record as an input A data base access system characterized by having the data-base-manipulation instruction conversion program which performs update operation of a virtual table record.

[Translation done.]

*** NOTICES ***

Japan Patent Office is not responsible for any damages caused by the use of this translation.

1.This document has been translated by computer. So the translation may not reflect the original precisely.

2.***** shows the word which can not be translated.

3.In the drawings, any words are not translated.

DETAILED DESCRIPTION

[Detailed Description of the Invention]

[0001]

[The technical field to which invention belongs] This invention relates to an access program strong against modification of a table structure which relates to the application program which uses the record of a data base, especially is made applicable to access.

[0002]

[Description of the Prior Art] According to JP,9-6801,A (data base access system), the data of various classes is shared in recent years, and the database management system for aiming at informational effective use is proposed. The Relational Database Management System which deals with data also in various kinds of database management systems by which the current proposal is made using the table in consideration of correlation of data has spread quickly.

[0003] The fundamental view of the data storage in a relational database is performed by using a table. Each element which a table is a 2-dimensional table which consists of a column (train) and a record (line), and constitutes a table is called column data. Moreover, since two or more columns (train) exist, one line, i.e., one record, is usually constituted by two or more kinds of column data.

[0004] The managerial system of such a relational database is a system indispensable to today's business society, and ISO (International Organization for Standardization) has specified the "SQL (Structured Query Language) functor" which is a relational database declinable word word, in order to define and operate a relational database.

[0005] According to the "HiRDB SQL reference" (, Inc. Hitachi), the "view" function is offered by the database management system. A view is the logical table which can access data from another table etc. In fact, the data itself is not necessarily stored in the view and the table used as the board of a view is called a "real table" to a view.

[0006] Even if it is data constituted based on two or more real tables in fact as an advantage which uses a view at the time of data base application system development, it can access as data of one table and there is no necessity that he is conscious of the complexity of an actual table structure. It is mentioned by renaming a train that the data seen from another view can be shown etc., without changing the definition of a real table in fact.

[0007] A view is defined by the SELECT sentence of SQL, the set of the retrieval activation result record of a SELECT sentence becomes one view as it is, and the column name of a view serves as an identifier specified at the time of the definition of a view.

[0008] When a view is used as a candidate for actuation of application system, it is necessary to correspond by changing the view definition of a database management system directly in the cases, such as an escape of a table, and modification the column data within application gives the significance [modification], and, according to "JP,9-237280,A" (data base retrieval equipment by ER model orientation), it is an activity done by an application programmer requesting a database manager.

[0009] Moreover, when the SELECT sentence of a definition of a view includes the conditions of "association", Update operation is Key-Preserved Table (all the keys of a real table). As opposed to the train of the real table in the case of holding the same key also with a view The possibility of, When only one Key-Preserved Table exists in a view, deletion actuation The possibility of, Insertion actuation is possible to the train of Key-Preserved Table, and when performing each actuation of updating, deletion, and insertion, the real table with which each actuation is reflected has a limit of being one.

[0010]

[Problem(s) to be Solved by the Invention] The data access in the alias name which carried out the virtual table of the train name of a real table by performing the view definition of a data base with the above-mentioned conventional technology is possible.

[0011] However, in the data base application system which refers to one data base (307) from two or more applications like especially drawing 3, when the form where an instruction is transmitted to a direct data base access interface (304,305,306) from an application program (301,302,303) is taken, the code of the application with which others are not changed, or the schema definition of a data base will be affected in the case of one or more application program modification.

[0012] When the code of application was changed, there is the necessity for recompilation, and big cost had to start, and it had to carry out by the application program having requested from the database manager modification of the data base schema definition including modification of a view definition of a database management system which is needed in the cases, such as an escape of a data base table, and modification within the application of the column data of a record the significance [modification] is given, and there was a problem that big cost started.

[0013] Moreover, when the SELECT sentence which defines a view includes the conditions of "association", Update operation only receives the train of Key-Preserved Table. The possibility of, Only when only one Key-Preserved Table exists in a view, deletion actuation The possibility of, Insertion actuation is possible only to the train of Key-Preserved Table. Although the real table with which each actuation is reflected has a limit of being one and a data base application developer can code in the reference actuation to a view, without being conscious of the structure of a real table when performing each actuation of updating, deletion, and insertion In each actuation of updating, deletion, and insertion, coating which was conscious of the structure of a real table had to be performed, and there was a problem that the consistency of actuation was missing.

[0014] This invention abolishes the necessity that he is conscious of the structure of a table also when performing updating and deletion / insertion actuation of a record at the time of application development. Reference actuation, It aims at giving consistency to the interface of updating and deletion / insertion actuation. Modification of the column data of the record furthermore used within application, It aims at abolishing the necessity for modification of the definition information in a database management system, and abolishing the necessity for code modification and recompilation of an application program in the case of modification and an escape of a table structure in the cases, such as modification column data gives the significance [modification].

[0015]

[Means for Solving the Problem] Since the above-mentioned technical problem is solved and the purpose of this invention is attained, the following means can be considered. A data base equipped with two or more real tables which are the sets of a real record which have one or more kinds of real column data, A database management means to acquire or change real column data of this data base, In a data base access system which consists of an access-control means to control acquisition or modification actuation of this real column data of said database management means, corresponding to a given SQL sentence Virtual table definition information which associates some real column data of one or more real tables of this data base as each virtual table column data of one virtual table, Virtual table operating instructions which acquire or change this virtual table column data are considered as an input. By generating one or more SQL sentences which operate it to real column data of one or more real tables of this data base, and giving this SQL sentence to this access-control means Acquisition or modification of real column data is operated. A result of acquisition of this real column data, or modification It is attained by offering a data base access system characterized by having the data-base-manipulation instruction transform-processing program outputted as acquisition or a modification result of virtual table column data.

[0016] Namely, this invention is set to data base application system using a relational database like drawing 4. A data base application program using data stored in a relational database (401,402,403), Between data base access interface programs (404,405,406) which offer access in an SQL sentence to a Relational Database Management System, one or more tables of a relational database A virtual table column name and table column name contrast information on a configuration of a virtual table definition information table (604) shown in drawing 6 acquired from 502 of a configuration of being shown in virtual table definition file (409,411,413) drawing 5 (501) of the program exterior (601) And it changes into a virtual table based on a

virtual table definition information table (414,415,416) which consists of joint column name set information (602) on a table, and record unique nature guarantee column name set information (603) in a table acquired from (503). Retrieval to this virtual table from an application program (401,402,403), Reflect a result of this actuation of insertion, modification, and an input of operating instructions of deletion to this virtual table. By changing into one or more SQL sentences which operate it to one or more tables, and publishing a generated SQL sentence to a data base access interface program (404,405,406) An instruction conversion program (408,410,412) which supposes a table structure as a virtual table to an application program (401,402,403) is arranged.

[0017] An instruction conversion program (408,410,412) receives virtual table record retrieval operating instructions. A virtual table column name train for retrieval, and information stored in a virtual table definition information table (414,415,416) from an input of retrieval conditions, From a virtual table column name which appears in input, a table name made applicable to retrieval is solved. A virtual table column name train for retrieval of an input, and a virtual table column name which appears in a retrieval condition It changes into a column name of a table. From virtual table definition information and a table name for retrieval A virtual table column name train for retrieval which generated a table connection condition and was changed into a column name of a table, Retrieval conditions changed into a column name of a table, a table name made applicable to retrieval, A retrieval instruction which generated a SELECT sentence of SQL and was generated from a table connection condition using a data base access interface of a relational database is published, and a column name of this retrieval result is changed and returned to a virtual table column name.

[0018] As opposed to virtual table record insertion operating instructions Moreover, information on a virtual table definition information table, From an input of inserted-record information which consists of a virtual table column name and a pair of train of column data Divide this inserted-record information per table for each insertion, and each virtual table column name of this table unit inserted record is changed into a column name of a table. To a table which serves as a candidate of each insertion contrast, generate a SELECT sentence of SQL of an adjustment check and adjustment after virtual table record insertion is checked. When adjustment is maintained, an INSERT sentence of SQL which generated an INSERT sentence of SQL for every table, and was generated using a data base access interface of this relational database is published.

[0019] Moreover, a DELETE sentence of SQL which generated deletion conditions which can guarantee data consistency after deletion to be virtual table definition information from an input of deletion conditions about all groups of a table which constitutes a virtual table to virtual table record deletion operating instructions, generated a DELETE sentence of SQL for every table based on enumerated deletion conditions, and was generated using a data base access interface of this relational database is published.

[0020] Moreover, virtual table update record information which consists of a group of virtual table definition information, a virtual table column name, and virtual table column data to a virtual table record update operation instruction, Virtual table record retrieval actuation which makes all virtual table column names a virtual table all column name for retrieval, and makes updating conditions as a retrieval condition input from updating conditions is performed. Virtual table column data of virtual table update record information replaces data of a virtual table column contained in virtual table update record information on a retrieval result virtual table record, and said virtual table record insertion actuation is performed by considering this after [substitute] retrieval result virtual table record as an input.

[0021]

[Embodiment of the Invention] Hereafter, the gestalt of operation of this invention is explained to details.

[0022] Drawing 1 is the block diagram of the data base application system which applied this invention, drawing 2 is flow drawing showing the flow of the whole processing to the instruction from an application program, drawing 6 is the block diagram of a virtual table definition file, drawing 7 is one example of the table format of the data base made applicable [target] to actuation, and drawing 8 is one example of the virtual table definition file which constitutes a virtual table for the table format of drawing 7 .

[0023] In drawing 1 , a server machine (115) consists of CPU (102), a keyboard (103), external storage (104), a communication device (105), main storage (107), and a bus (106), stores a relational database (RDB) (120) in external storage (104), and loads and performs a Relational Database Management System (RDBMS) (122) on main storage (107).

[0024] Moreover, a client machine (116) consists of CPU (109), a keyboard (111), external storage (110), a

communication device (108), main storage (114), and a bus (113), stores a virtual table definition file (121) on external storage (110), and loads and performs a data base application program (AP), (117), an instruction conversion program (123), and a virtual table setting program (119) on main storage (114). An instruction conversion program (123) reads the information on a virtual table definition file (121) into a virtual table definition information table (118) from external storage (110).

[0025] Virtual table definition information is stored in the virtual table definition information table (604) which has the structure shown by drawing 6 from the virtual table definition file (501) of a configuration of that drawing 5 shows. "The virtual table column name and table column name contrast information, and table joint column name set information (502)" on a virtual table definition file (501) are stored also in the joint column name information (602) on a table, when it stores in the virtual table column name and table column name contrast information (601) on a virtual table definition information table (604) and the column name of two or more tables is equivalent to one virtual table column name. Moreover (501) stores the record unique nature guarantee column name set information (503) in a table in the record unique nature guarantee column name set information (603) in a table on a virtual table definition information table (604).

[0026] In drawing 2, an instruction conversion program (202) reads into a virtual table definition information table (208) the information on the virtual table definition file (203) beforehand created using the virtual table setting program (204). If AP (201) publishes an instruction of retrieval actuation of a virtual table record, insertion actuation, deletion actuation, or update operation to an instruction conversion program (202) Based on this virtual table definition information, these virtual table record operating instructions The SELECT sentence of one or more SQL, Or it changes into the combination of an INSERT sentence or a DELETE sentence, publishes to the access interface (207) of the target relational database, and is operated to RDB (206) through RDBMS (205).

[0027] The processing to which an instruction conversion program (202) performs hereafter the table format shown in drawing 7 to retrieval of a virtual table record, insertion, deletion, and an updating instruction for an example is explained.

[0028] Drawing 8 (801) is the example of the virtual table definition file for constituting a virtual table V (704) from a table A (701), a table B (702), and a table C (703), and drawing 9 (704) is a virtual table definition information table which reads and generates a virtual table definition file (801).

[0029] Drawing 10 is the block diagram of an instruction conversion program. The case where the virtual table column name train (1201) for retrieval and virtual table record retrieval conditions (1202) are hereafter published for the virtual table record retrieval instruction (1200) shown in drawing 12 as an input to the motion to virtual table record retrieval processing instruction of instruction conversion program instruction-conversion program (1020) for the virtual table V (704) is made into an example, and it explains.

[0030] Drawing 11 is flow drawing showing the processing to a retrieval instruction. In (step 1100), the virtual table definition information reading section (1016) reads a virtual table definition file (1017), and stores it in a virtual table definition information table (1015). At (step 1101), the instruction reception section (1100) receives the input of the virtual table column name train (1201) for retrieval, and virtual table record retrieval conditions (1202) as a retrieval instruction.

[0031] At (step 1102), the virtual table column name train (1201) for retrieval received at (step 1101) is passed to the virtual table column name processing section (1002) for retrieval, and virtual table record retrieval conditions (1202) are passed to the retrieval condition processing section (1003), respectively. The virtual table column name processing section (1002) for retrieval and the retrieval condition processing section (1003) register into the column name Management Department (1012) for retrieval the virtual table column name which appears during each input here. The virtual table column name registered in the input of (1200) here registers all the appearance virtual table column names that are the passages of (1203), and control moves from it to the SQL generation section (1001) immediately after the end.

[0032] At (step 1103), the SQL generation section (1001) requests solution of the table name for retrieval from the table name for retrieval, and the column name decision section (1004). In (step 1103), it is decided in the input of (1200) that it will be the table (1204) which serves as a candidate for retrieval from (1204).

[0033] At (step 1104), the SQL generation section calls the virtual table column name processing section (1002) for retrieval, and the virtual table record retrieval condition processing section (1003), and acquires the column name train for retrieval and retrieval conditions after column name conversion. At this time, the virtual table column name processing section (1002) for retrieval and the virtual table record retrieval

condition processing section (1003) call the table name for retrieval, and the column name decision section (1004), acquire the table name made applicable to retrieval, and change a column name. It is made for the column name of a retrieval result to turn into a virtual table column name about the column name for retrieval here by surrounding with a double quotation mark after each column name for retrieval. the input of (1200) — receiving — each of (1206) — the output of the column name (1207) for retrieval and retrieval conditions (1208) is obtained.

[0034] At (step 1105), the SQL generation section calls the table connection condition generation section (1005), and acquires the connection condition about the table used as the candidate for retrieval. In the table connection condition generation section (1005), the table name for retrieval is acquired from the table name for retrieval, and the column name decision section (1004), and a table connection condition is generated based on the table joint column name information on a virtual table definition information table (1015). (1209) will be generated to the input of (1200).

[0035] At (step 1106), the SELECT sentence of SQL is generated based on the table name for retrieval generated at (step 1103), the column name for retrieval generated at (step 1104), retrieval conditions, and the table connection condition generated at (step 1105). To the input of (1200), if (1210), it is generated.

[0036] At (step 1107), the SQL sentence generated in the SQL instruction generation section (1001) in the SQL instruction issue section (1010) is published to a data base access interface (1018). At (step 1108), it changes into the format which uses a retrieval result with application again in the retrieval result processing section (1011).

[0037] Next, the case where a virtual table column name (1401) and virtual table column data (1402) are published for the virtual table record insertion instruction (1400) shown in drawing 14 as an input to the motion to virtual table record insertion processing instruction of instruction conversion program instruction-conversion program (1020) for the virtual table V (704) is made into an example, and it explains.

[0038] Drawing 13 is flow drawing showing the processing to an insertion instruction. In (step 1300), the virtual table definition information reading section (1016) reads a virtual table definition file (1017), and stores it in a virtual table definition information table (1015). At (step 1301), the instruction reception section (1100) receives the input of the virtual table record insertion instruction (1400) which serves as a virtual table column name (1401) from virtual table column data (1402) as an insertion instruction.

[0039] At (step 1302), in the virtual table record division processing section (1006), a virtual table record is divided for every table corresponding to each column, and it registers with an inserted-record information management table (1013).

[0040] When a virtual table record insertion instruction (1400) is an input here, it becomes as shown in (1406) and they are the portion (1403) corresponding to Table A, a portion (1404) corresponding to Table B, and a portion (1405) corresponding to Table C, respectively. It sets up that it is unsettled about Table A, Table B, and Table C here.

[0041] A transaction is started at (step 1303). This is because virtual table record insertion actuation is offered as one actuation to the exterior of an instruction conversion program to consisting of one or more data variation orders on SQL level, so it is necessary to consider all data base modification processings in an instruction conversion program as one ATOMIKKU actuation in all.

[0042] At (step 1304), it investigates whether there is any table which is a candidate for data insertion and is not processed yet with reference to the inserted-record information management table (1013) in the insertion adjustment check section (1007). In the example of drawing 14, since Table A, Table B, and Table C are unsettled, it is YES.

[0043] At (step 1305), it is a table for insertion, and one unsettled table is chosen and it is referred to as T. In the example of drawing 14, it is arbitrary and Table A is set to T. At (step 1306), it inspects whether the record unique nature guarantee column in a table is shown in the table chosen at (step 1305) with reference to a virtual table definition information table. On the table A of an example, since ID column is a unique nature guarantee column, it is set to YES here.

[0044] It searches with (step 1309) whether the unique nature guarantee column of T exists in the inserted-record information about T altogether. In an example, the unique nature guarantee column of A is only ID, will exist altogether in the inserted-record information (1403) about A, and serves as YES. Since it becomes impossible to offer the adjustment guarantee at the time of inserting when it does not exist, the roll back of the transaction is carried out (step 1314), and it ends.

[0045] It searches with (step 1310) whether the data of the unique nature guarantee column of the inserted-record information about T and the record of equivalence exist. At this time, SQL used for inspection is a SELECT sentence with the conditional expression with which only the number of columns connected the conditional expression of the [column name = column data] of inserted-record information on AND conditions, and let the columns for retrieval be all columns. In the case of the example in drawing 14, checking SQL is set to (1407).

[0046] Since a retrieval result serves as a record whose number may not have been zero here, a record may be inserted (step 1307). Inserted-record information is compared with the record of a result searched with (step 1312) in (step 1311). Since it means that a mismatch occurs when it differs, the roll back of the transaction is carried out (step 1314), and it ends. When equivalent, it means that the same record already exists and it is not necessary to insert a record.

[0047] Since the record and inserted-record information on the result searched with the example are equivalent, it turns out that it is not necessary to insert a record in Table A. At (step 1308), T is set as a processing settled. It carries out about Table B and Table C like the following. Consequently, about Table B, 1409 and the SQL sentence of 1410 are generated about 1408 and Table C. After processing of Table B and Table C is completed, in branching of (step 1304), it is set to NO, and it commits (step 1313) and a transaction is ended.

[0048] Next, the case where the virtual table record deletion instruction (1700) shown in drawing 17 is published as a virtual table record deletion condition input to the motion to the virtual table record deletion instruction of instruction conversion program instruction-conversion program (1020) for the virtual table V (704) is made into an example, and it explains.

[0049] Drawing 15 is flow drawing showing the processing to a deletion instruction. In (step 1500), the virtual table definition information reading section (1016) reads a virtual table definition file (1017), and stores it in a virtual table definition information table (1015). At (step 1522), the instruction reception section (1100) receives the input of virtual table record deletion conditions as a virtual table record deletion instruction. In the example of drawing 17, (1700) is considered as a virtual table record deletion condition input.

[0050] At (step 1501), it divides for every table related in the virtual table record deletion conditions of an input in the virtual table record condition processing section (1008), and stores in a deletion condition information management table (1014). A deletion condition managed table (1014) consists of a deletion condition table (1600) with NxN structure, and a root flag table (1601) with 1xN structure here, when the number of tables contained in a virtual table is set to N, as shown in drawing 16.

[0051] In (step 1501), the conditions which divided the virtual table record deletion conditions of an input for every table are stored in the location of (T, T) of a deletion condition table (1600), when a corresponding table is set to T, respectively. A root flag table (1601) is initialized at OFF. (1701) shows the contents of the deletion condition table (1600) in a termination (step 1501) time to the input of (1700), and (1704) shows the contents of the root flag table.

[0052] A transaction is started at (step 1521). This is because virtual table record deletion actuation is offered as one actuation to the exterior of an instruction conversion program to consisting of one or more data variation orders on SQL level, so it is necessary to consider all data base modification processings in an instruction conversion program as one ATOMIKKU actuation in all.

[0053] At (step 1502), it inspects whether the table on which the element with which a root flag table (1601) corresponds serves as OFF exists. When it does not exist, it progresses to (step 1509). At (step 1503), a root flag chooses one of the arbitration from the tables of OFF, considers as the root, and sets the element of a corresponding root flag table (1601) to ON. Moreover, a current flag table is initialized at OFF.

[0054] At (step 1504), deletion conditions get down from close to the column with which the line corresponding to the root table of a deletion condition table (1600) corresponds, and it inspects whether the table whose current flag is OFF exists. It inspects whether the table on which conditions are not in close exists in the column with which one is chosen from them and current ** and a current flag are set when it exists (step 1505), and it is the candidate for joint of KARENTO, and the line corresponding to the root of a deletion condition table (1600) corresponds (step 1506).

[0055] When it does not exist (step 1504). When it exists, the table of the corresponding arbitration is set as a join, a search which makes the joint column of KARENTO and a join the train for retrieval a condition

[the (current ** root) of a deletion condition table (1600)] is performed (step 1508), and it inspects whether a retrieval result exists (step 1519). When it does not exist, the roll back of the transaction is carried out (step 1520), and it ends.

[0056] When it exists, about a retrieval result record, the number of duplications is counted, respectively (step 1518), retrieval which makes retrieval conditions that with which only the number of record columns combined the conditions of "column name = column data" format by AND association based on the retrieval result record, and uses the column for retrieval as all columns is performed about each record, and the number of hits is counted (step 1517).

[0057] It inspects whether the record the number of duplications computed at (step 1518) here and whose number of hits called for at (step 1517) correspond exists (step 1515). When it does not exist, a deletion impossible flag is set as the join of a deletion condition table (1600), and the root (step 1516), and it progresses to (step 1506). When it exists, the conditions which generated the conditions with which only the number of record columns combined the conditions of "column name = column data" format by AND association about each record in agreement, and combined the each generated conditions by OR association are stored in (the join and the root) of a deletion condition table (1600) (step 1514), and it progresses to (step 1506).

[0058] (1702) is the contents of the deletion condition table constituted to the input of (1700) here. Moreover (1705), they are the contents of the root flag table in the time of (1702) being constituted. In (step 1502), when the table of OFF of the value of a root flag table does not exist, conditional expression of each line is published about each train of a deletion condition table (1600) to the table which joins together and corresponds by AND association (1513).

[0059] It is not operated about the table on which the deletion impossible flag is set up or close one is not [conditions] in the train to which a deletion condition table corresponds here (step 1512). (1703) is conditions of the deletion to each table generated to the input of (1700) here. It commits (step 1510) and a transaction is ended in the place which processing of all tables finished.

[0060] Next, the motion to the virtual table record update process instruction of an instruction conversion program is explained.

[0061] Drawing 18 is flow drawing showing the processing to an updating instruction, and drawing 19 is the example of the renewal instruction of a virtual table record. The virtual table definition information reading section (1016) reads a virtual table definition file (1017), and stores it in a virtual table definition information table (1015) (step 1800). The instruction reception section (1100) receives the input of the renewal instruction (1900) of a virtual table record which serves as a virtual table column name (1901) and virtual table column data (1902) from the renewal conditions (1903) of a virtual table record as an updating instruction (step 1801).

[0062] A transaction is started at (step 1810). This is because virtual table record update operation provides as one actuation to the exterior of an instruction conversion program to consisting of one or more data variation orders, so it needs to consider all data base modification processings in an instruction conversion program as one ATOMIKKU actuation in all on SQL level.

[0063] At (step 1802), all virtual table column names are specified as a virtual table record retrieval condition input as the renewal conditions of a virtual table record, and a virtual table column name for retrieval, and virtual table record retrieval processing is performed. A retrieval result virtual table record inspects one or more affairs of a certain thing (step 1803), as having no updating object record, in the case of zero affair, the roll back of the transaction is carried out (step 1808), and it is completed.

[0064] When a retrieval result exists, virtual table record deletion is performed as a virtual table record deletion condition input (step 1804). At (step 1805), it inspects whether a retrieval result set at (step 1802) is empty. When it is empty, it commits (step 1809) and a transaction is ended.

[0065] When it is not empty, the column data corresponding to the column name included in ejection and virtual table update record information in one of the arbitration is changed into the column data of virtual table update record information (step 1806), and virtual table record insertion actuation is performed as a virtual table record which inserts the changed virtual table record (step 1807), and it progresses to (step 1805).

[0066] According to the above-mentioned example, it is effective in becoming possible to change, without the executor of application changing the definition of a database management system for the object table of application.

[0067] In the above-mentioned example, the specification method of the target virtual table [conversion program / instruction] can consider the following methods.

[0068] Drawing 20 is the example of a configuration of the application with which application changes the target data at the assignment at the time of activation and which uses an instruction conversion program. External storage (2003) stores the execution file (2004) of the application which uses an instruction conversion program, a merchandise management virtual table definition file (2005), and a customer management virtual table definition file (2006).

[0069] When a goods data base application activation means (2001) is performed, (2004) is loaded to main storage (2007), stores the information on a merchandise management virtual table definition file (2005) in a merchandise management virtual table definition information table (2010), and becomes the goods data base application (2008) which operates a goods data control data base (2012).

[0070] When a customer database application activation means (2002) is performed, (2004) is loaded to main storage (2007), stores the information on a customer management virtual table definition file (2006) in a customer management virtual table definition information table (2011), and becomes the customer database application (2009) which operates a customer data control data base (2013).

[0071] In the above-mentioned example, it becomes possible for an application program to be able to change the target virtual table with an easy means to specify a definition file name at the time of starting, for example, to use the same application program of logic as another application like the goods data base application of the above-mentioned example, and customer database application.

[0072] Moreover, when [this] the virtual table operating instructions in the above-mentioned example may be SQL to a virtual table, the inside of the FROM phrase of a SELECT sentence and a DELETE sentence and the INTO phrase of an INSERT sentence serves as a virtual table name.

[0073] In this example, even when SQL functor is extended, the effect by the virtual table can be realized.

[0074] For example, the example of the following escapes can be considered.

[0075] a SELECT news number, F1 (news, the 'report . text', 'report [the text {"Hitachi"}]'), and price FROM news table WHERE F — F1 and F2 are extended syntax 2 (news, 'report [the text {"Hitachi"}]') IS TRUE here.

[0076] In this case, the virtual table column name for retrieval in a previous example corresponds to NewsNo and F1(News, the 'report . text', 'report [the text {"Hitachi"}]') VALUE, virtual table record retrieval conditions correspond to F2(News, 'report [the text {"Hitachi"}]') IS TRUE, and the FROM phrase corresponding to the real table for retrieval and a connection condition are generated.

[0077] An output is SELECT NewsMaster.NewsNo [which is as follows], F1 (NewsMaster.News. 'the report . text', 'report [the text {"Hitachi"}]'), News Price.PriceFROM NewsMaster, and NewsPriceWHERE F2 (NewsMaster.News, 'report [the text {"Hitachi"}]') ISTRUE AND NewsMaster.NewsNo=NewsPrice.NewsNo [0078].

[Effect of the Invention] As stated above, since a data base application developer can perform coding which was conscious only of the actuation to one table supposed by the configuration file definition, without being conscious of the complicated structure of data base table substance, according to this invention, he has the effect of data base application man day reduction.

[0079] Moreover, since no necessity for modification of the code of a program or recompilation is and it can respond only by Make Changes of data-base access a configuration file also when semantic attachment within the definition of data-base table substance and the application of the column data of a record changed into it, since the name with which a definition was supposed is used into the data-base access a configuration file of the program exterior by the column name for actuation in a program code, the effect of maintenance-cost reduction is.

[Translation done.]

*** NOTICES ***

Japan Patent Office is not responsible for any damages caused by the use of this translation.

1.This document has been translated by computer. So the translation may not reflect the original precisely.

2.**** shows the word which can not be translated.

3.In the drawings, any words are not translated.

TECHNICAL FIELD

[The technical field to which invention belongs] This invention relates to an access program strong against modification of a table structure which relates to the application program which uses the record of a data base, especially is made applicable to access.

[Translation done.]

*** NOTICES ***

Japan Patent Office is not responsible for any damages caused by the use of this translation.

1.This document has been translated by computer. So the translation may not reflect the original precisely.

2.**** shows the word which can not be translated.

3.In the drawings, any words are not translated.

PRIOR ART

[Description of the Prior Art] According to JP,9-6801,A (data base access system), the data of various classes is shared in recent years, and the database management system for aiming at informational effective use is proposed. The Relational Database Management System which deals with data also in various kinds of database management systems by which the current proposal is made using the table in consideration of correlation of data has spread quickly.

[0003] The fundamental view of the data storage in a relational database is performed by using a table. Each element which a table is a 2-dimensional table which consists of a column (train) and a record (line), and constitutes a table is called column data. Moreover, since two or more columns (train) exist, one line, i.e., one record, is usually constituted by two or more kinds of column data.

[0004] The managerial system of such a relational database is a system indispensable to today's business society, and ISO (International Organization for Standardization) has specified the "SQL (Structured Query Language) functor" which is a relational database declinable word word, in order to define and operate a relational database.

[0005] According to the "HiRDB SQL reference" (, Inc. Hitachi), the "view" function is offered by the database management system. A view is the logical table which can access data from another table etc. In fact, the data itself is not necessarily stored in the view and the table used as the board of a view is called a "real table" to a view.

[0006] Even if it is data constituted based on two or more real tables in fact as an advantage which uses a view at the time of data base application system development, it can access as data of one table and there is no necessity that he is conscious of the complexity of an actual table structure. It is mentioned by renaming a train that the data seen from another view can be shown etc., without changing the definition of a real table in fact.

[0007] A view is defined by the SELECT sentence of SQL, the set of the retrieval activation result record of a SELECT sentence becomes one view as it is, and the column name of a view serves as an identifier specified at the time of the definition of a view.

[0008] When a view is used as a candidate for actuation of application system, it is necessary to correspond by changing the view definition of a database management system directly in the cases, such as an escape of a table, and modification the column data within application gives the significance [modification], and, according to "JP,9-237280,A" (data base retrieval equipment by ER model orientation), it is an activity done by an application programmer requesting a database manager.

[0009] Moreover, when the SELECT sentence of a definition of a view includes the conditions of "association", update operation should receive the train of Key-Preserved Table (real table in case all the keys of a real table hold the same key also with a view). When, as for the ** possibility of and deletion actuation, only one Key-Preserved Table exists in a view, the possibility of and insertion actuation are possible to the train of Key-Preserved Table, and when performing each actuation of updating, deletion, and insertion, the real table with which each actuation is reflected has a limit of being one.

[Translation done.]

*** NOTICES ***

Japan Patent Office is not responsible for any damages caused by the use of this translation.

1.This document has been translated by computer. So the translation may not reflect the original precisely.

2.**** shows the word which can not be translated.

3.In the drawings, any words are not translated.

EFFECT OF THE INVENTION

[Effect of the Invention] As stated above, since a data base application developer can perform coding which was conscious only of the actuation to one table supposed by the configuration file definition, without being conscious of the complicated structure of data base table substance, according to this invention, he has the effect of data base application man day reduction.

[0079] Moreover, since no necessity for modification of the code of a program or recompilation is and it can respond only by Make Changes of data-base access a configuration file also when semantic attachment within the definition of data-base table substance and the application of the column data of a record changed into it, since the name with which a definition was supposed is used into the data-base access a configuration file of the program exterior by the column name for actuation in a program code, the effect of maintenance-cost reduction is.

[Translation done.]

* NOTICES *

Japan Patent Office is not responsible for any damages caused by the use of this translation.

1.This document has been translated by computer. So the translation may not reflect the original precisely.

2.***** shows the word which can not be translated.

3.In the drawings, any words are not translated.

TECHNICAL PROBLEM

[Problem(s) to be Solved by the Invention] The data access in the alias name which carried out the virtual table of the train name of a real table by performing the view definition of a data base with the above-mentioned conventional technology is possible.

[0011] However, in the data base application system which refers to one data base (307) from two or more applications like especially drawing 3 , when the form where an instruction is transmitted to a direct data base access interface (304,305,306) from an application program (301,302,303) is taken, the code of the application with which others are not changed, or the schema definition of a data base will be affected in the case of one or more application program modification.

[0012] When the code of application was changed, there is the necessity for recompilation, and big cost had to start, and it had to carry out by the application program having requested from the database manager modification of the data base schema definition including modification of a view definition of a database management system which is needed in the cases, such as an escape of a data base table, and modification within the application of the column data of a record the significance [modification] is given, and there was a problem that big cost started.

[0013] Moreover, when the SELECT sentence which defines a view includes the conditions of "association", Update operation only receives the train of Key-Preserved Table. The possibility of, Only when only one Key-Preserved Table exists in a view, deletion actuation The possibility of, Insertion actuation is possible only to the train of Key-Preserved Table. Although the real table with which each actuation is reflected has a limit of being one and a data base application developer can code in the reference actuation to a view, without being conscious of the structure of a real table when performing each actuation of updating, deletion, and insertion In each actuation of updating, deletion, and insertion, coating which was conscious of the structure of a real table had to be performed, and there was a problem that the consistency of actuation was missing.

[0014] This invention abolishes the necessity that he is conscious of the structure of a table also when performing updating and deletion / insertion actuation of a record at the time of application development. Reference actuation, It aims at giving consistency to the interface of updating and deletion / insertion actuation. Modification of the column data of the record furthermore used within application, It aims at abolishing the necessity for modification of the definition information in a database management system, and abolishing the necessity for code modification and recompilation of an application program in the case of modification and an escape of a table structure in the cases, such as modification column data gives the significance [modification].

[Translation done.]

*** NOTICES ***

Japan Patent Office is not responsible for any damages caused by the use of this translation.

1.This document has been translated by computer. So the translation may not reflect the original precisely.

2.**** shows the word which can not be translated.

3.In the drawings, any words are not translated.

MEANS

[Means for Solving the Problem] Since the above-mentioned technical problem is solved and the purpose of this invention is attained, the following means can be considered. A data base equipped with two or more real tables which are the sets of a real record which have one or more kinds of real column data, A database management means to acquire or change real column data of this data base, In a data base access system which consists of an access-control means to control acquisition or modification actuation of this real column data of said database management means, corresponding to a given SQL sentence Virtual table definition information which associates some real column data of one or more real tables of this data base as each virtual table column data of one virtual table, Virtual table operating instructions which acquire or change this virtual table column data are considered as an input. By generating one or more SQL sentences which operate it to real column data of one or more real tables of this data base, and giving this SQL sentence to this access-control means Acquisition or modification of real column data is operated. A result of acquisition of this real column data, or modification It is attained by offering a data base access system characterized by having the data-base-manipulation instruction transform-processing program outputted as acquisition or a modification result of virtual table column data.

[0016] Namely, this invention is set to data base application system using a relational database like drawing 4 . A data base application program using data stored in a relational database (401,402,403), Between data base access interface programs (404,405,406) which offer access in an SQL sentence to a Relational Database Management System, one or more tables of a relational database A virtual table column name and table column name contrast information on a configuration of a virtual table definition information table (604) shown in drawing 6 acquired from 502 of a configuration of being shown in virtual table definition file (409,411,413) drawing 5 (501) of the program exterior (601) And it changes into a virtual table based on a virtual table definition information table (414,415,416) which consists of joint column name set information (602) on a table, and record unique nature guarantee column name set information (603) in a table acquired from (503). Retrieval to this virtual table from an application program (401,402,403), Reflect a result of this actuation of insertion, modification, and an input of operating instructions of deletion to this virtual table. By changing into one or more SQL sentences which operate it to one or more tables, and publishing a generated SQL sentence to a data base access interface program (404,405,406) An instruction conversion program (408,410,412) which supposes a table structure as a virtual table to an application program (401,402,403) is arranged.

[0017] An instruction conversion program (408,410,412) receives virtual table record retrieval operating instructions. A virtual table column name train for retrieval, and information stored in a virtual table definition information table (414,415,416) from an input of retrieval conditions, From a virtual table column name which appears in input, a table name made applicable to retrieval is solved. A virtual table column name train for retrieval of an input, and a virtual table column name which appears in a retrieval condition It changes into a column name of a table. From virtual table definition information and a table name for retrieval A virtual table column name train for retrieval which generated a table connection condition and was changed into a column name of a table, Retrieval conditions changed into a column name of a table, a table name made applicable to retrieval, A retrieval instruction which generated a SELECT sentence of SQL and was generated from a table connection condition using a data base access interface of a relational database is published, and a column name of this retrieval result is changed and returned to a virtual table column name.

[0018] As opposed to virtual table record insertion operating instructions Moreover, information on a

virtual table definition information table, From an input of inserted-record information which consists of a virtual table column name and a pair of train of column data Divide this inserted-record information per table for each insertion, and each virtual table column name of this table unit inserted record is changed into a column name of a table. To a table which serves as a candidate of each insertion contrast, generate a SELECT sentence of SQL of an adjustment check and adjustment after virtual table record insertion is checked. When adjustment is maintained, an INSERT sentence of SQL which generated an INSERT sentence of SQL for every table, and was generated using a data base access interface of this relational database is published.

[0019] Moreover, a DELETE sentence of SQL which generated deletion conditions which can guarantee data consistency after deletion to be virtual table definition information from an input of deletion conditions about all groups of a table which constitutes a virtual table to virtual table record deletion operating instructions, generated a DELETE sentence of SQL for every table based on enumerated deletion conditions, and was generated using a data base access interface of this relational database is published.

[0020] Moreover, virtual table update record information which consists of a group of virtual table definition information, a virtual table column name, and virtual table column data to a virtual table record update operation instruction, Virtual table record retrieval actuation which makes all virtual table column names a virtual table all column name for retrieval, and makes updating conditions as a retrieval condition input from updating conditions is performed. Virtual table column data of virtual table update record information replaces data of a virtual table column contained in virtual table update record information on a retrieval result virtual table record, and said virtual table record insertion actuation is performed by considering this after [substitute] retrieval result virtual table record as an input.

[0021]

[Embodiment of the Invention] Hereafter, the gestalt of operation of this invention is explained to details.

[0022] Drawing 1 is the block diagram of the data base application system which applied this invention, drawing 2 is flow drawing showing the flow of the whole processing to the instruction from an application program, drawing 6 is the block diagram of a virtual table definition file, drawing 7 is one example of the table format of the data base made applicable [target] to actuation, and drawing 8 is one example of the virtual table definition file which constitutes a virtual table for the table format of drawing 7.

[0023] In drawing 1, a server machine (115) consists of CPU (102), a keyboard (103), external storage (104), a communication device (105), main storage (107), and a bus (106), stores a relational database (RDB) (120) in external storage (104), and loads and performs a Relational Database Management System (RDBMS) (122) on main storage (107).

[0024] Moreover, a client machine (116) consists of CPU (109), a keyboard (111), external storage (110), a communication device (108), main storage (114), and a bus (113), stores a virtual table definition file (121) on external storage (110), and loads and performs a data base application program (AP), (117), an instruction conversion program (123), and a virtual table setting program (119) on main storage (114). An instruction conversion program (123) reads the information on a virtual table definition file (121) into a virtual table definition information table (118) from external storage (110).

[0025] Virtual table definition information is stored in the virtual table definition information table (604) which has the structure shown by drawing 6 from the virtual table definition file (501) of a configuration of that drawing 5 shows. "The virtual table column name and table column name contrast information, and table joint column name set information (502)" on a virtual table definition file (501) are stored also in the joint column name information (602) on a table, when it stores in the virtual table column name and table column name contrast information (601) on a virtual table definition information table (604) and the column name of two or more tables is equivalent to one virtual table column name. Moreover (501) stores the record unique nature guarantee column name set information (503) in a table in the record unique nature guarantee column name set information (603) in a table on a virtual table definition information table (604).

[0026] In drawing 2, an instruction conversion program (202) reads into a virtual table definition information table (208) the information on the virtual table definition file (203) beforehand created using the virtual table setting program (204). If AP (201) publishes an instruction of retrieval actuation of a virtual table record, insertion actuation, deletion actuation, or update operation to an instruction conversion program (202) Based on this virtual table definition information, these virtual table record operating instructions The SELECT sentence of one or more SQL, Or it changes into the combination of an INSERT sentence or a DELETE sentence, publishes to the access interface (207) of the target relational database,

and is operated to RDB (206) through RDBMS (205).

[0027] The processing to which an instruction conversion program (202) performs hereafter the table format shown in drawing 7 to retrieval of a virtual table record, insertion, deletion, and an updating instruction for an example is explained.

[0028] Drawing 8 (801) is the example of the virtual table definition file for constituting a virtual table V (704) from a table A (701), a table B (702), and a table C (703), and drawing 9 (704) is a virtual table definition information table which reads and generates a virtual table definition file (801).

[0029] Drawing 10 is the block diagram of an instruction conversion program. The case where the virtual table column name train (1201) for retrieval and virtual table record retrieval conditions (1202) are hereafter published for the virtual table record retrieval instruction (1200) shown in drawing 12 as an input to the motion to virtual table record retrieval processing instruction of instruction conversion program instruction-conversion program (1020) for the virtual table V (704) is made into an example, and it explains.

[0030] Drawing 11 is flow drawing showing the processing to a retrieval instruction. In (step 1100), the virtual table definition information reading section (1016) reads a virtual table definition file (1017), and stores it in a virtual table definition information table (1015). At (step 1101), the instruction reception section (1100) receives the input of the virtual table column name train (1201) for retrieval, and virtual table record retrieval conditions (1202) as a retrieval instruction.

[0031] At (step 1102), the virtual table column name train (1201) for retrieval received at (step 1101) is passed to the virtual table column name processing section (1002) for retrieval, and virtual table record retrieval conditions (1202) are passed to the retrieval condition processing section (1003), respectively. The virtual table column name processing section (1002) for retrieval and the retrieval condition processing section (1003) register into the column name Management Department (1012) for retrieval the virtual table column name which appears during each input here. The virtual table column name registered in the input of (1200) here registers all the appearance virtual table column names that are the passages of (1203), and control moves from it to the SQL generation section (1001) immediately after the end.

[0032] At (step 1103), the SQL generation section (1001) requests solution of the table name for retrieval from the table name for retrieval, and the column name decision section (1004). In (step 1103), it is decided in the input of (1200) that it will be the table (1204) which serves as a candidate for retrieval from (1204).

[0033] At (step 1104), the SQL generation section calls the virtual table column name processing section (1002) for retrieval, and the virtual table record retrieval condition processing section (1003), and acquires the column name train for retrieval and retrieval conditions after column name conversion. At this time, the virtual table column name processing section (1002) for retrieval and the virtual table record retrieval condition processing section (1003) call the table name for retrieval, and the column name decision section (1004), acquire the table name made applicable to retrieval, and change a column name. It is made for the column name of a retrieval result to turn into a virtual table column name about the column name for retrieval here by surrounding with a double quotation mark after each column name for retrieval. the input of (1200) — receiving — each of (1206) — the output of the column name (1207) for retrieval and retrieval conditions (1208) is obtained.

[0034] At (step 1105), the SQL generation section calls the table connection condition generation section (1005), and acquires the connection condition about the table used as the candidate for retrieval. In the table connection condition generation section (1005), the table name for retrieval is acquired from the table name for retrieval, and the column name decision section (1004), and a table connection condition is generated based on the table joint column name information on a virtual table definition information table (1015). (1209) will be generated to the input of (1200).

[0035] At (step 1106), the SELECT sentence of SQL is generated based on the table name for retrieval generated at (step 1103), the column name for retrieval generated at (step 1104), retrieval conditions, and the table connection condition generated at (step 1105). To the input of (1200), if (1210), it is generated.

[0036] At (step 1107), the SQL sentence generated in the SQL instruction generation section (1001) in the SQL instruction issue section (1010) is published to a data base access interface (1018). At (step 1108), it changes into the format which uses a retrieval result with application again in the retrieval result processing section (1011).

[0037] Next, the case where a virtual table column name (1401) and virtual table column data (1402) are published for the virtual table record insertion instruction (1400) shown in drawing 14 as an input to the

motion to virtual table record insertion processing instruction of instruction conversion program instruction-conversion program (1020) for the virtual table V (704) is made into an example, and it explains.

[0038] Drawing 13 is flow drawing showing the processing to an insertion instruction. In (step 1300), the virtual table definition information reading section (1016) reads a virtual table definition file (1017), and stores it in a virtual table definition information table (1015). At (step 1301), the instruction reception section (1100) receives the input of the virtual table record insertion instruction (1400) which serves as a virtual table column name (1401) from virtual table column data (1402) as an insertion instruction.

[0039] At (step 1302), in the virtual table record division processing section (1006), a virtual table record is divided for every table corresponding to each column, and it registers with an inserted-record information management table (1013).

[0040] When a virtual table record insertion instruction (1400) is an input here, it becomes as shown in (1406) and they are the portion (1403) corresponding to Table A, a portion (1404) corresponding to Table B, and a portion (1405) corresponding to Table C, respectively. It sets up that it is unsettled about Table A, Table B, and Table C here.

[0041] A transaction is started at (step 1303). This is because virtual table record insertion actuation is offered as one actuation to the exterior of an instruction conversion program to consisting of one or more data variation orders on SQL level, so it is necessary to consider all data base modification processings in an instruction conversion program as one ATOMIKKU actuation in all.

[0042] At (step 1304), it investigates whether there is any table which is a candidate for data insertion and is not processed yet with reference to the inserted-record information management table (1013) in the insertion adjustment check section (1007). In the example of drawing 14, since Table A, Table B, and Table C are unsettled, it is YES.

[0043] At (step 1305), it is a table for insertion, and one unsettled table is chosen and it is referred to as T. In the example of drawing 14, it is arbitrary and Table A is set to T. At (step 1306), it inspects whether the record unique nature guarantee column in a table is shown in the table chosen at (step 1305) with reference to a virtual table definition information table. On the table A of an example, since ID column is a unique nature guarantee column, it is set to YES here.

[0044] It searches with (step 1309) whether the unique nature guarantee column of T exists in the inserted-record information about T altogether. In an example, the unique nature guarantee column of A is only ID, will exist altogether in the inserted-record information (1403) about A, and serves as YES. Since it becomes impossible to offer the adjustment guarantee at the time of inserting when it does not exist, the roll back of the transaction is carried out (step 1314), and it ends.

[0045] It searches with (step 1310) whether the data of the unique nature guarantee column of the inserted-record information about T and the record of equivalence exist. At this time, SQL used for inspection is a SELECT sentence with the conditional expression with which only the number of columns connected the conditional expression of the [column name = column data] of inserted-record information on AND conditions, and let the columns for retrieval be all columns. In the case of the example in drawing 14, checking SQL is set to (1407).

[0046] Since a retrieval result serves as a record whose number may not have been zero here, a record may be inserted (step 1307). Inserted-record information is compared with the record of a result searched with (step 1312) in (step 1311). Since it means that a mismatch occurs when it differs, the roll back of the transaction is carried out (step 1314), and it ends. When equivalent, it means that the same record already exists and it is not necessary to insert a record.

[0047] Since the record and inserted-record information on the result searched with the example are equivalent, it turns out that it is not necessary to insert a record in Table A. At (step 1308), T is set as a processing settled. It carries out about Table B and Table C like the following. Consequently, about Table B, 1409 and the SQL sentence of 1410 are generated about 1408 and Table C. After processing of Table B and Table C is completed, in branching of (step 1304), it is set to NO, and it commits (step 1313) and a transaction is ended.

[0048] Next, the case where the virtual table record deletion instruction (1700) shown in drawing 17 is published as a virtual table record deletion condition input to the motion to the virtual table record deletion instruction of instruction conversion program instruction-conversion program (1020) for the virtual table V (704) is made into an example, and it explains.

[0049] Drawing 15 is flow drawing showing the processing to a deletion instruction. In (step 1500), the virtual table definition information reading section (1016) reads a virtual table definition file (1017), and stores it in a virtual table definition information table (1015). At (step 1522), the instruction reception section (1100) receives the input of virtual table record deletion conditions as a virtual table record deletion instruction. In the example of drawing 17, (1700) is considered as a virtual table record deletion condition input.

[0050] At (step 1501), it divides for every table related in the virtual table record deletion conditions of an input in the virtual table record condition processing section (1008), and stores in a deletion condition information management table (1014). A deletion condition managed table (1014) consists of a deletion condition table (1600) with NxN structure, and a root flag table (1601) with 1xN structure here, when the number of tables contained in a virtual table is set to N, as shown in drawing 16.

[0051] In (step 1501), the conditions which divided the virtual table record deletion conditions of an input for every table are stored in the location of (T, T) of a deletion condition table (1600), when a corresponding table is set to T, respectively. A root flag table (1601) is initialized at OFF. (1701) shows the contents of the deletion condition table (1600) in a termination (step 1501) time to the input of (1700), and (1704) shows the contents of the root flag table.

[0052] A transaction is started at (step 1521). This is because virtual table record deletion actuation is offered as one actuation to the exterior of an instruction conversion program to consisting of one or more data variation orders on SQL level, so it is necessary to consider all data base modification processings in an instruction conversion program as one ATOMIKKU actuation in all.

[0053] At (step 1502), it inspects whether the table on which the element with which a root flag table (1601) corresponds serves as OFF exists. When it does not exist, it progresses to (step 1509). At (step 1503), a root flag chooses one of the arbitration from the tables of OFF, considers as the root, and sets the element of a corresponding root flag table (1601) to ON. Moreover, a current flag table is initialized at OFF.

[0054] At (step 1504), deletion conditions get down from close to the column with which the line corresponding to the root table of a deletion condition table (1600) corresponds, and it inspects whether the table whose current flag is OFF exists. It inspects whether the table on which conditions are not in close exists in the column with which one is chosen from them and current ** and a current flag are set when it exists (step 1505), and it is the candidate for joint of KARENTO, and the line corresponding to the root of a deletion condition table (1600) corresponds (step 1506).

[0055] When it does not exist (step 1504). When it exists, the table of the corresponding arbitration is set as a join, a search which makes the joint column of KARENTO and a join the train for retrieval a condition [the (current ** root) of a deletion condition table (1600)] is performed (step 1508), and it inspects whether a retrieval result exists (step 1519). When it does not exist, the roll back of the transaction is carried out (step 1520), and it ends.

[0056] When it exists, about a retrieval result record, the number of duplications is counted, respectively (step 1518), retrieval which makes retrieval conditions that with which only the number of record columns combined the conditions of "column name = column data" format by AND association based on the retrieval result record, and uses the column for retrieval as all columns is performed about each record, and the number of hits is counted (step 1517).

[0057] It inspects whether the record the number of duplications computed at (step 1518) here and whose number of hits called for at (step 1517) correspond exists (step 1515). When it does not exist, a deletion impossible flag is set as the join of a deletion condition table (1600), and the root (step 1516), and it progresses to (step 1506). When it exists, the conditions which generated the conditions with which only the number of record columns combined the conditions of "column name = column data" format by AND association about each record in agreement, and combined the each generated conditions by OR association are stored in (the join and the root) of a deletion condition table (1600) (step 1514), and it progresses to (step 1506).

[0058] (1702) is the contents of the deletion condition table constituted to the input of (1700) here. Moreover (1705), they are the contents of the root flag table in the time of (1702) being constituted. In (step 1502), when the table of OFF of the value of a root flag table does not exist, conditional expression of each line is published about each train of a deletion condition table (1600) to the table which joins together and corresponds by AND association (1513).

[0059] It is not operated about the table on which the deletion impossible flag is set up or close one is not [conditions] in the train to which a deletion condition table corresponds here (step 1512). (1703) is conditions of the deletion to each table generated to the input of (1700) here. It commits (step 1510) and a transaction is ended in the place which processing of all tables finished.

[0060] Next, the motion to the virtual table record update process instruction of an instruction conversion program is explained.

[0061] Drawing 18 is flow drawing showing the processing to an updating instruction, and drawing 19 is the example of the renewal instruction of a virtual table record. The virtual table definition information reading section (1016) reads a virtual table definition file (1017), and stores it in a virtual table definition information table (1015) (step 1800). The instruction reception section (1100) receives the input of the renewal instruction (1900) of a virtual table record which serves as a virtual table column name (1901) and virtual table column data (1902) from the renewal conditions (1903) of a virtual table record as an updating instruction (step 1801).

[0062] A transaction is started at (step 1810). This is because virtual table record update operation provides as one actuation to the exterior of an instruction conversion program to consisting of one or more data variation orders, so it needs to consider all data base modification processings in an instruction conversion program as one ATOMIKKU actuation in all on SQL level.

[0063] At (step 1802), all virtual table column names are specified as a virtual table record retrieval condition input as the renewal conditions of a virtual table record, and a virtual table column name for retrieval, and virtual table record retrieval processing is performed. A retrieval result virtual table record inspects one or more affairs of a certain thing (step 1803), as having no updating object record, in the case of zero affair, the roll back of the transaction is carried out (step 1808), and it is completed.

[0064] When a retrieval result exists, virtual table record deletion is performed as a virtual table record deletion condition input (step 1804). At (step 1805), it inspects whether a retrieval result set at (step 1802) is empty. When it is empty, it commits (step 1809) and a transaction is ended.

[0065] When it is not empty, the column data corresponding to the column name included in ejection and virtual table update record information in one of the arbitration is changed into the column data of virtual table update record information (step 1806), and virtual table record insertion actuation is performed as a virtual table record which inserts the changed virtual table record (step 1807), and it progresses to (step 1805).

[0066] According to the above-mentioned example, it is effective in becoming possible to change, without the executor of application changing the definition of a database management system for the object table of application.

[0067] In the above-mentioned example, the specification method of the target virtual table [conversion program / instruction] can consider the following methods.

[0068] Drawing 20 is the example of a configuration of the application with which application changes the target data at the assignment at the time of activation and which uses an instruction conversion program. External storage (2003) stores the execution file (2004) of the application which uses an instruction conversion program, a merchandise management virtual table definition file (2005), and a customer management virtual table definition file (2006).

[0069] When a goods data base application activation means (2001) is performed, (2004) is loaded to main storage (2007), stores the information on a merchandise management virtual table definition file (2005) in a merchandise management virtual table definition information table (2010), and becomes the goods data base application (2008) which operates a goods data control data base (2012).

[0070] When a customer database application activation means (2002) is performed, (2004) is loaded to main storage (2007), stores the information on a customer management virtual table definition file (2006) in a customer management virtual table definition information table (2011), and becomes the customer database application (2009) which operates a customer data control data base (2013).

[0071] In the above-mentioned example, it becomes possible for an application program to be able to change the target virtual table with an easy means to specify a definition file name at the time of starting, for example, to use the same application program of logic as another application like the goods data base application of the above-mentioned example, and customer database application.

[0072] Moreover, when [this] the virtual table operating instructions in the above-mentioned example may be SQL to a virtual table, the inside of the FROM phrase of a SELECT sentence and a DELETE

sentence and the INTO phrase of an INSERT sentence serves as a virtual table name.

[0073] In this example, even when SQL functor is extended, the effect by the virtual table can be realized.

[0074] For example, the example of the following escapes can be considered.

[0075] a SELECT news number, F1 (news, the 'report . text', 'report [the text {"Hitachi"}]'), and price FROM news table WHERE F — F1 and F2 are extended syntax 2 (news, 'report [the text {"Hitachi"}]') IS TRUE here.

[0076] In this case, the virtual table column name for retrieval in a previous example corresponds to NewsNo and F1(News, the 'report . text', 'report [the text {"Hitachi"}]') VALUE, virtual table record retrieval conditions correspond to F2(News, 'report [the text {"Hitachi"}]') IS TRUE, and the FROM phrase corresponding to the real table for retrieval and a connection condition are generated.

[0077] An output is SELECT NewsMaster.NewsNo [which is as follows], F1 (NewsMaster.News. 'the report . text', 'report [the text {"Hitachi"}]'), News Price.PriceFROM NewsMaster, and NewsPriceWHERE F2 (NewsMaster.News, 'report [the text {"Hitachi"}]') ISTRUE AND NewsMaster.NewsNo=NewsPrice.NewsNo.

[Translation done.]

* NOTICES *

Japan Patent Office is not responsible for any damages caused by the use of this translation.

1.This document has been translated by computer. So the translation may not reflect the original precisely.

2.**** shows the word which can not be translated.

3.In the drawings, any words are not translated.

DESCRIPTION OF DRAWINGS

[Brief Description of the Drawings]

[Drawing 1] It is the block diagram of the data base application system which applied this invention.

[Drawing 2] It is flow drawing showing the flow of the whole processing to the instruction from an application program.

[Drawing 3] It is the block diagram of the data base application system which refers to one data base from two or more applications.

[Drawing 4] It is the block diagram of the data base application system which applied this invention which refers to one data base from two or more applications.

[Drawing 5] It is the block diagram of a virtual table definition file.

[Drawing 6] It is the block diagram of a virtual table definition information table.

[Drawing 7] It is table format drawing of the data base used as a candidate for actuation with the gestalt of operation.

[Drawing 8] It is drawing showing one example of the virtual table definition file which constitutes a virtual table for the table format of drawing 7.

[Drawing 9] It is drawing showing the virtual table definition information table which reads and generates the virtual table definition file of drawing 8.

[Drawing 10] It is the block diagram showing the configuration of an instruction conversion program.

[Drawing 11] It is flow drawing showing the processing to a retrieval instruction.

[Drawing 12] It is flow drawing showing the example of the data about a retrieval instruction.

[Drawing 13] It is flow drawing showing the processing to an insertion instruction.

[Drawing 14] It is flow drawing showing the example of the data about an insertion instruction.

[Drawing 15] It is flow drawing showing the processing to a deletion instruction.

[Drawing 16] It is a block diagram showing the configuration of a deletion condition managed table.

[Drawing 17] It is the example of a virtual table record deletion instruction.

[Drawing 18] It is flow drawing showing the processing to an updating instruction.

[Drawing 19] It is the block diagram showing the example of the renewal instruction of a virtual table record.

[Drawing 20] It is drawing showing the example of a configuration of the application with which application uses the instruction conversion program which changes the target data by assignment at the time of activation.

[Description of Notations]

201 — Application program 202 — An instruction conversion program, 203 — Virtual table definition file, 204 — A virtual table setting program, 205 — Database management equipment, 206 — A data base, 207 — Access interface 208 — A virtual table definition information table, 1000 — Instruction reception section, 1001 — The SQL generation section, 1002 — The virtual table column name processing section for retrieval, 1003 — The virtual table record retrieval condition processing section, 1004 — The table name for retrieval, and the column name decision section, 1005 — The table connection condition generation section, 1006 — Virtual table record division processing section 1007 [— Deletion adjustment guarantee section,] — The insertion adjustment check section, 1008 — The virtual table record condition division processing section, 1009 1010 — The SQL instruction issue section, 1011 — Retrieval result processing section, 1012 — The virtual table for retrieval, 1013 — The column name Management Department, 1014 — Inserted-record information management table, 1015 — A virtual table definition

information table, 1016 — Virtual table definition information reading section 1017 [— Instruction conversion program.] — A virtual table definition file, 1018 — A data base access interface, 1019

[Translation done.]

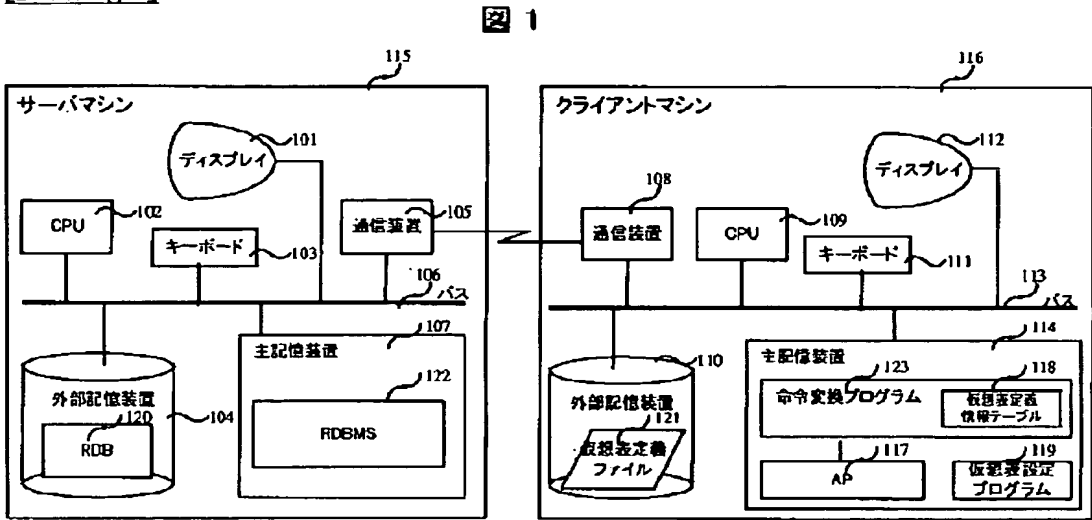
* NOTICES *

Japan Patent Office is not responsible for any damages caused by the use of this translation.

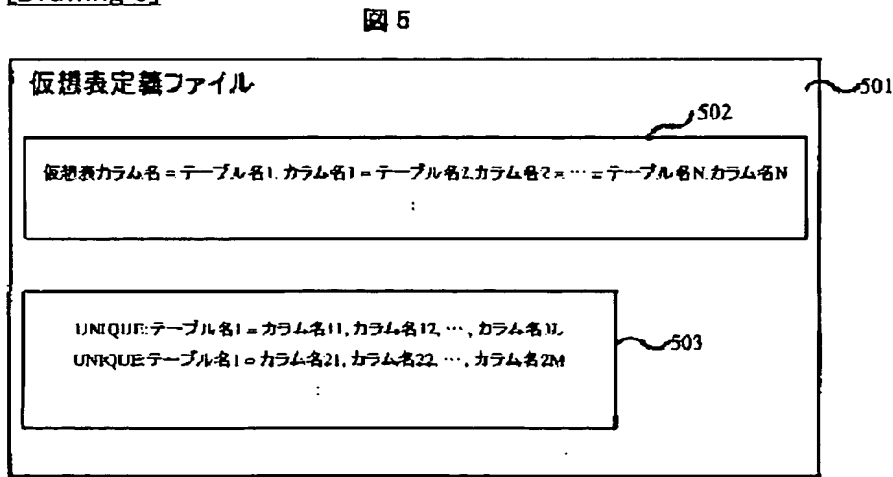
- 1.This document has been translated by computer. So the translation may not reflect the original precisely.
- 2.**** shows the word which can not be translated.
- 3.In the drawings, any words are not translated.

DRAWINGS

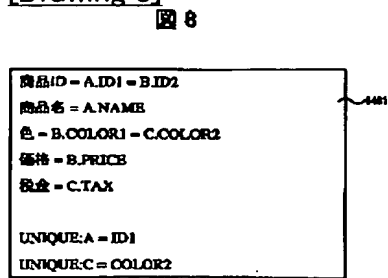
[Drawing 1]



[Drawing 5]

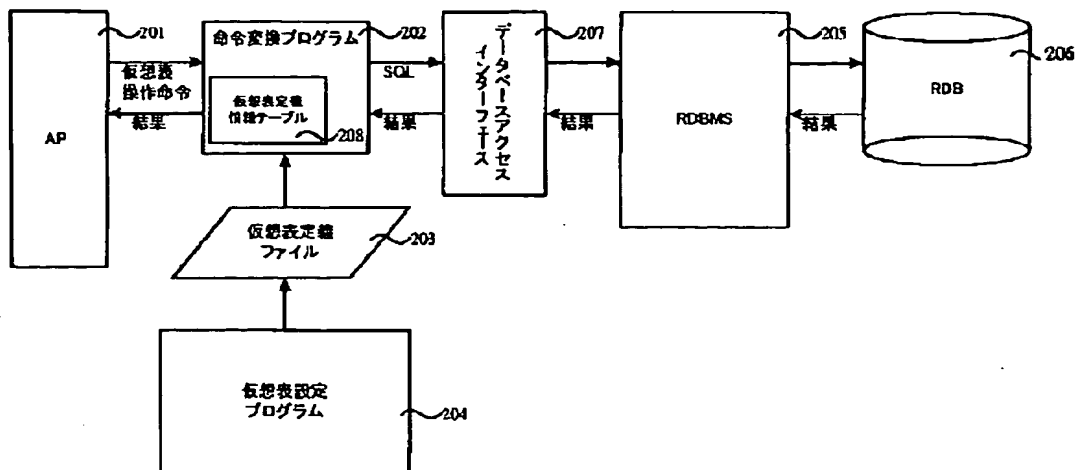


[Drawing 8]



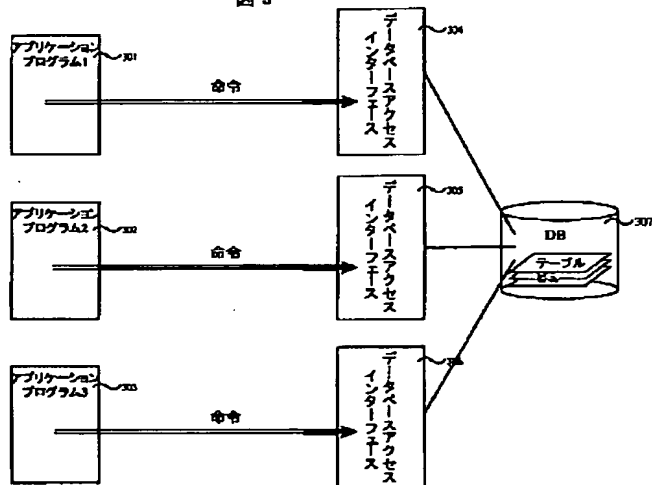
[Drawing 2]

図 2



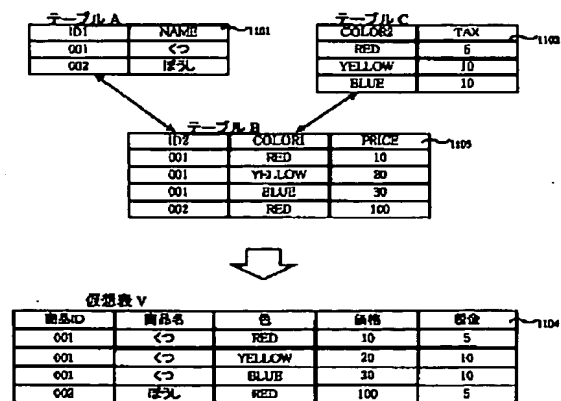
[Drawing 3]

図 3



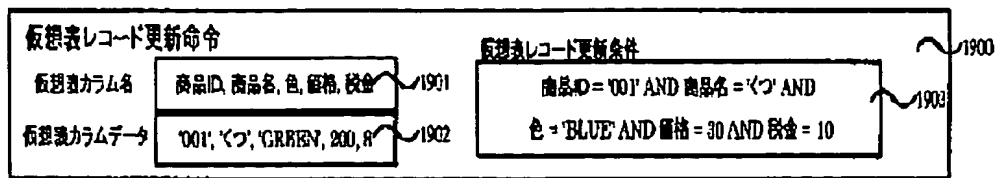
[Drawing 7]

図 7



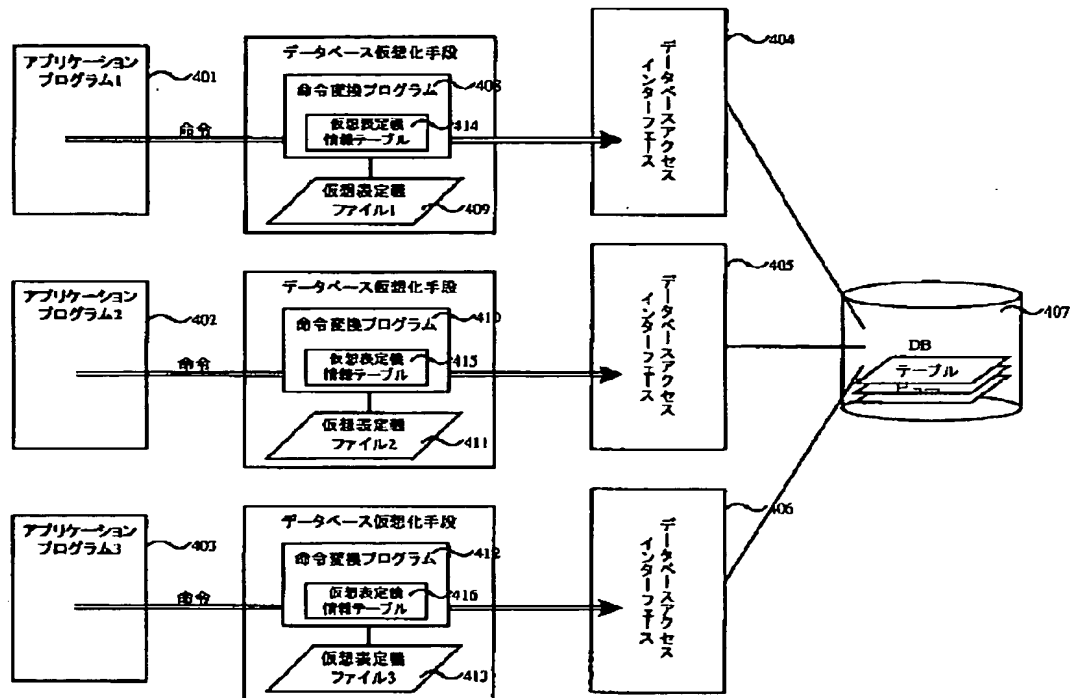
[Drawing 19]

図 19



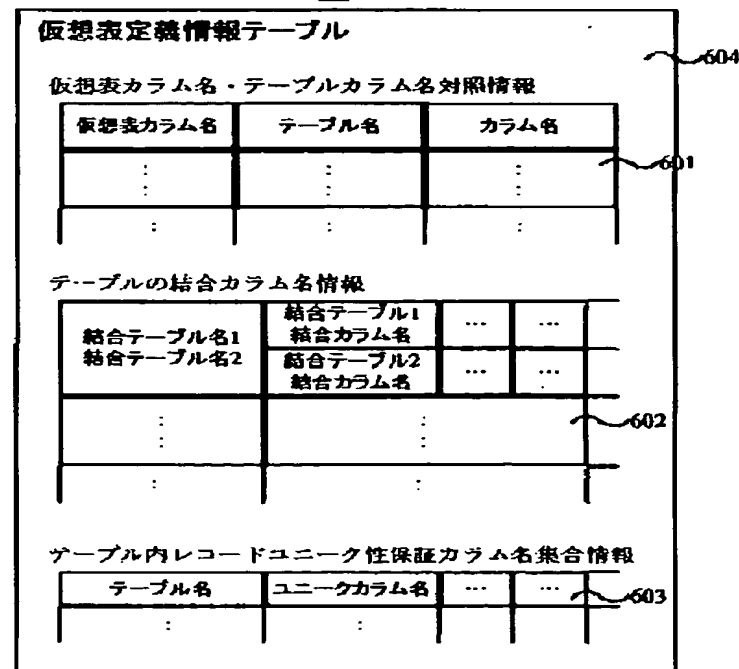
[Drawing 4]

図 4



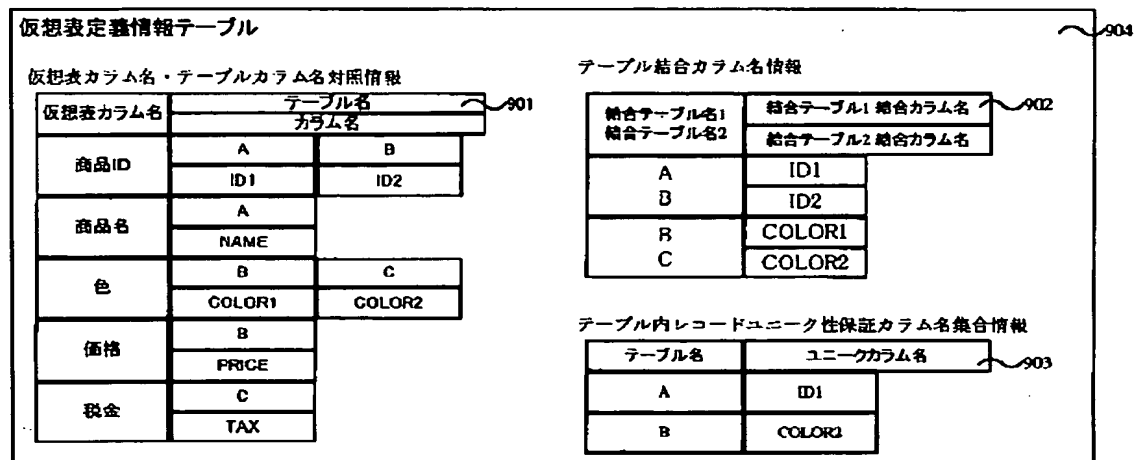
[Drawing 6]

図 6



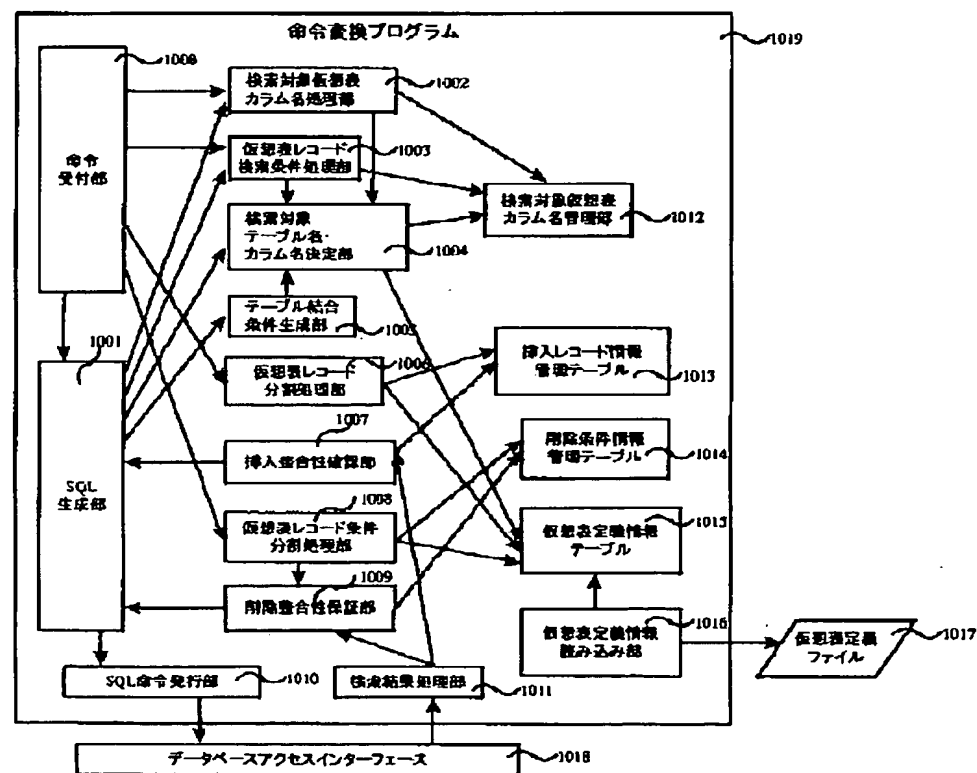
[Drawing 9]

図 9



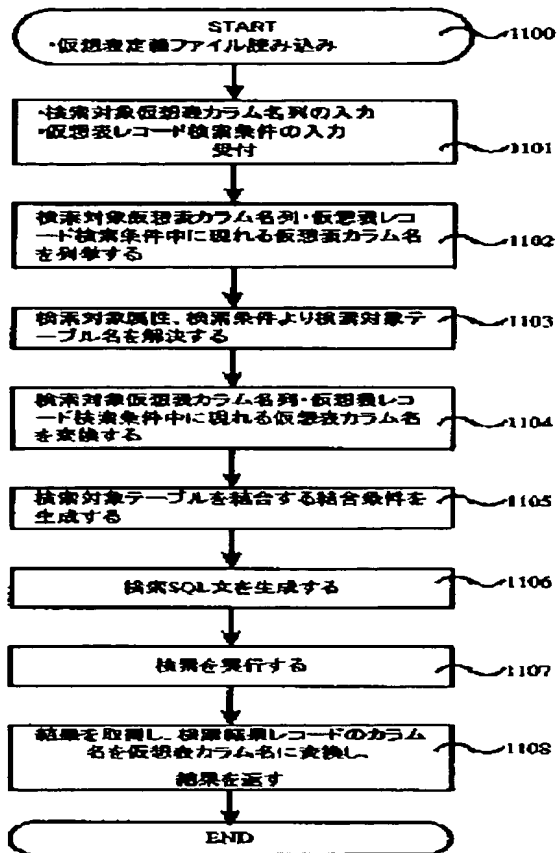
[Drawing 10]

図 10



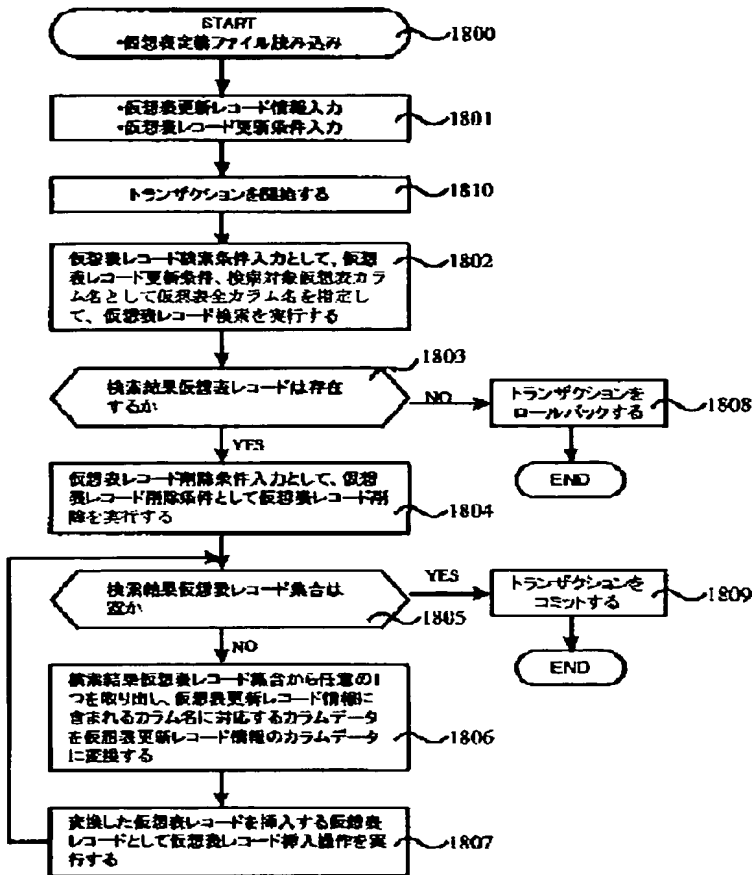
[Drawing 11]

図 11



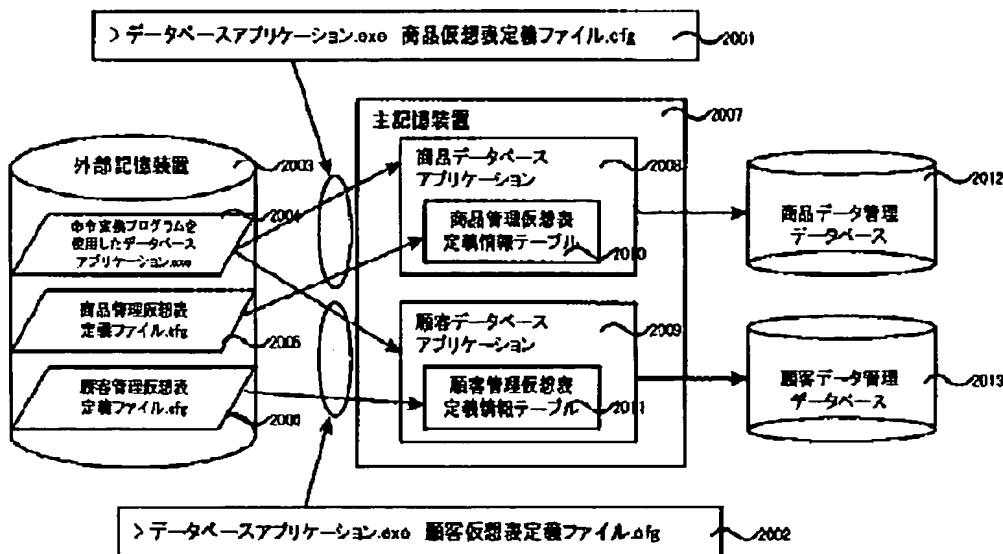
[Drawing 18]

図 18



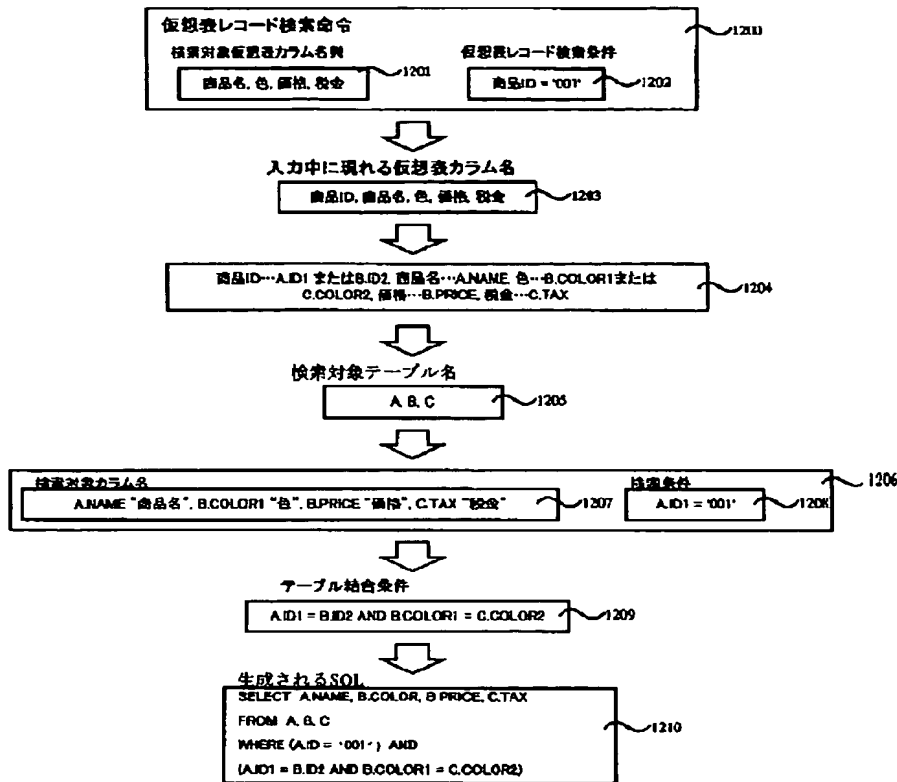
[Drawing 20]

図 20



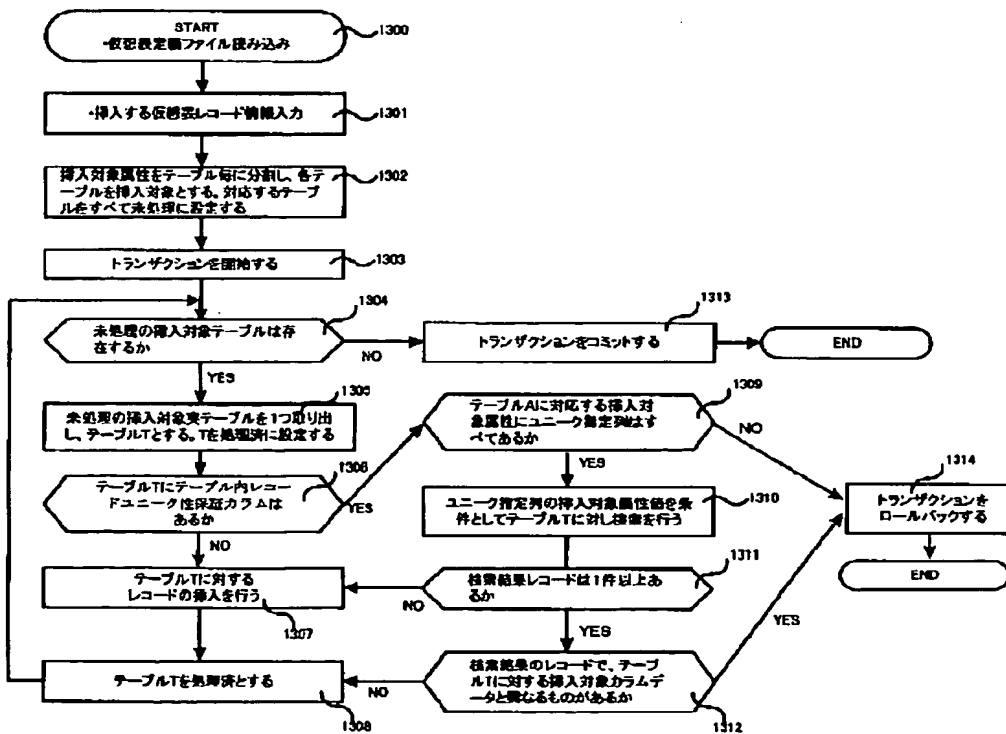
[Drawing 12]

図 12



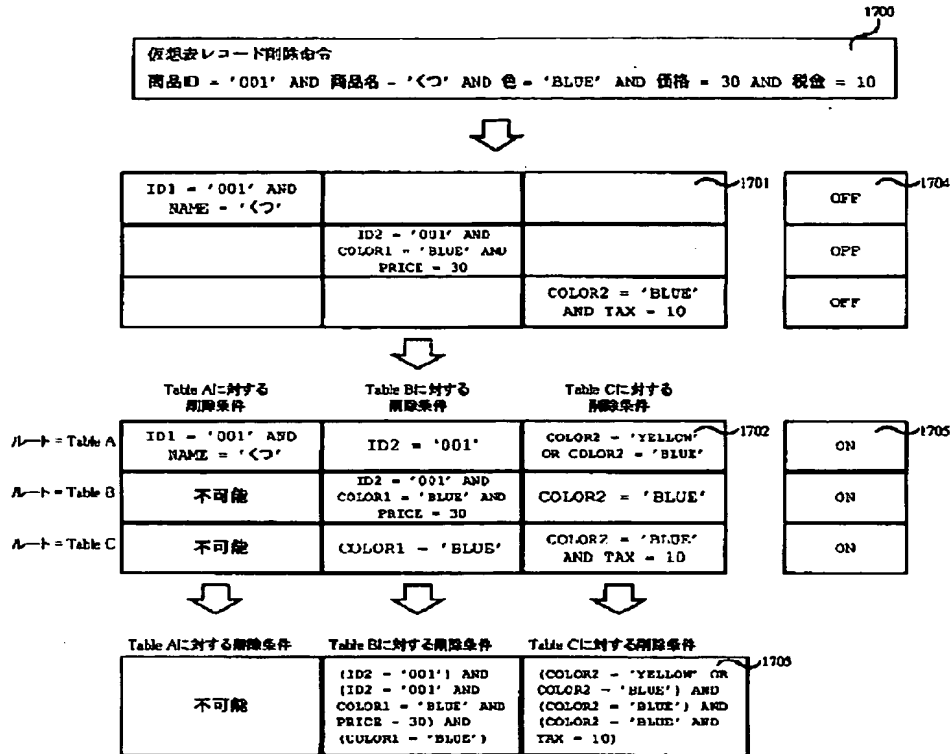
[Drawing 13]

図 13



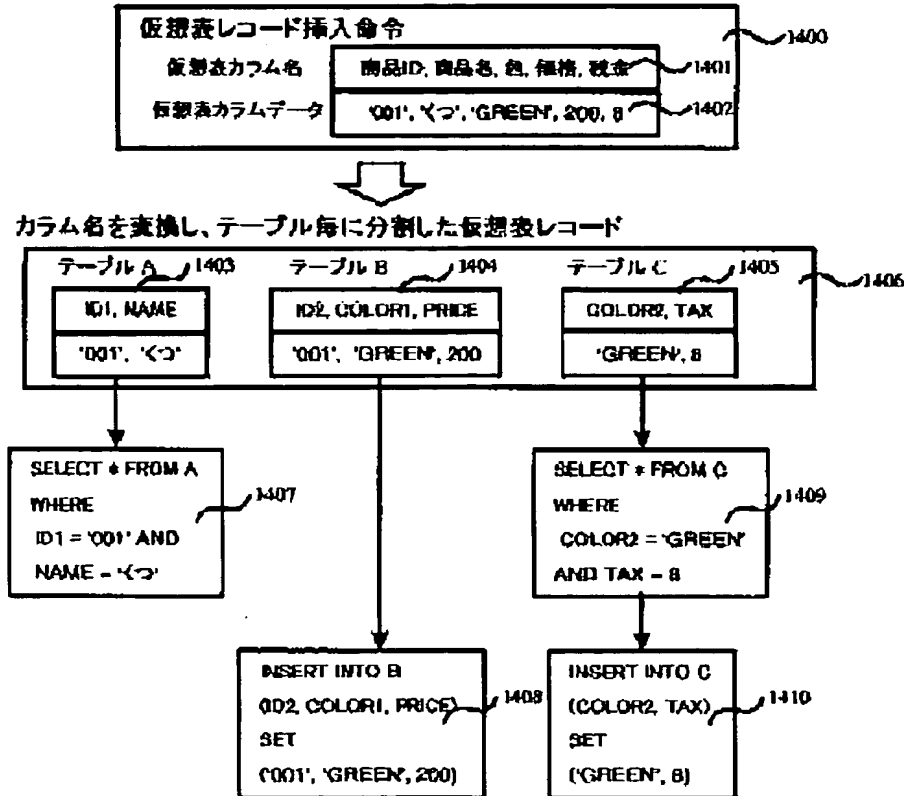
[Drawing 17]

図 17



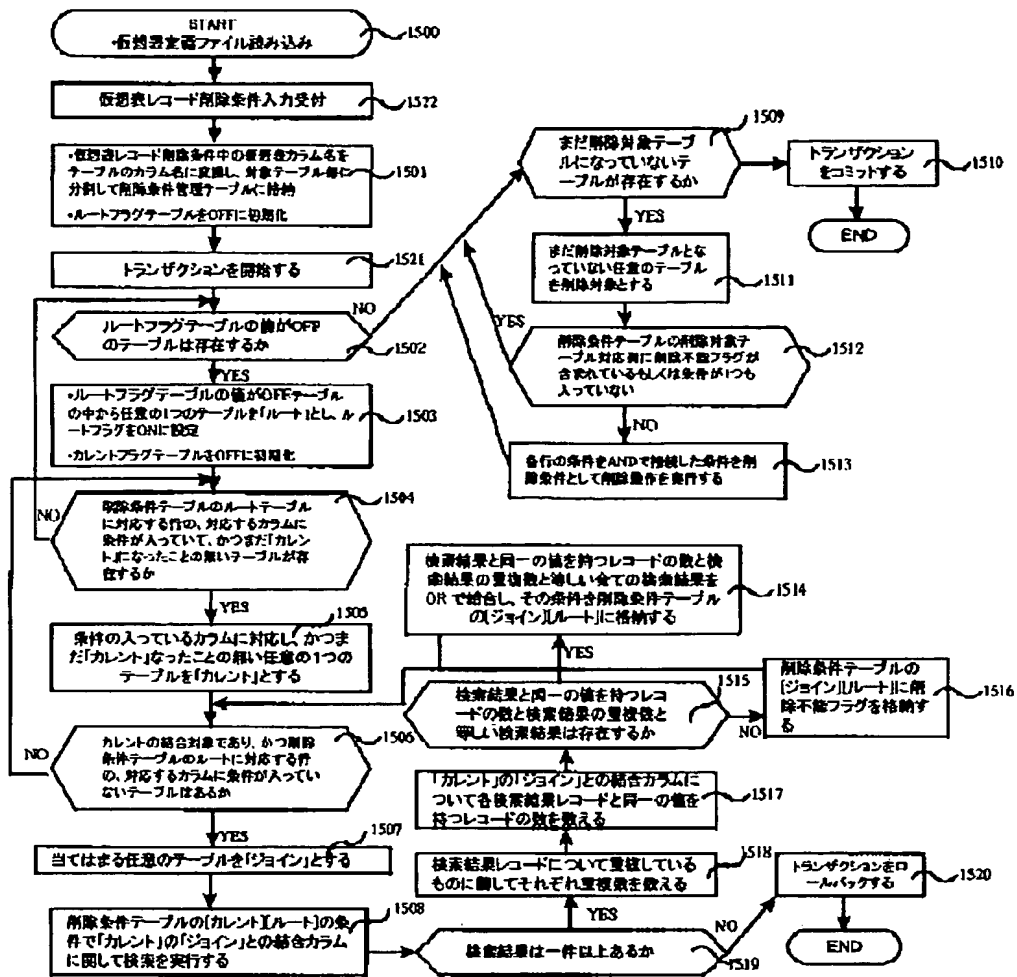
[Drawing 14]

図 14



[Drawing 15]

圖 15



[Drawing 16]

図 16

削除条件テーブル

	テーブル1に対する 削除条件	テーブル2に対する 削除条件	テーブル3に対する 削除条件	...
テーブル1をルートとしたときに導き出される 削除条件	仮想表レコード削除条件の テーブル1に関する テーブルレコード削除条件	テーブル1をルートとした ときに導き出されるテーブル 2に対する削除条件	テーブル1をルートとした ときに導き出されるテーブル 3に対する削除条件	... 1600
テーブル2をルートとし たときに導き出される 削除条件	テーブル2をルートとしたと きに導き出されるテーブル 1に対する削除条件	仮想表レコード削除条件の テーブル2に関する テーブルレコード削除条件	テーブル2をルートとしたと きに導き出されるテーブル 2に対する削除条件	...
テーブル3をルートとし たときに導き出される 削除条件	テーブル3をルートとしたと きに導き出されるテーブル 1に対する削除条件	テーブル3をルートとしたと きに導き出されるテーブル 2に対する削除条件	仮想表レコード削除条件の テーブル3に関する テーブルレコード削除条件	...
⋮	⋮	⋮	⋮	

ルートフラグテーブル

テーブル1	テーブル1がルートとなっ たことがあるか否かを示す [ON/OFF]
テーブル2	テーブル2がルートとなっ たことがあるか否かを示す [ON/OFF]
テーブル3	テーブル3がルートとなっ たことがあるか否かを示す [ON/OFF]
⋮	⋮

カレントフラグテーブル

テーブル1	テーブル1がカレントとなっ たことがあるか否かを示す [ON/OFF]
テーブル2	テーブル2がカレントとなっ たことがあるか否かを示す [ON/OFF]
テーブル3	テーブル3がカレントとなっ たことがあるか否かを示す [ON/OFF]
⋮	⋮

[Translation done.]

* NOTICES *

Japan Patent Office is not responsible for any damages caused by the use of this translation.

1.This document has been translated by computer. So the translation may not reflect the original precisely.

2.**** shows the word which can not be translated.

3.In the drawings, any words are not translated.

PRIOR ART

[Description of the Prior Art] The data of a data base is stored in the logical schema section and the live-data section on a hard disk drive unit, and DBMS which offers the development environment of database system will mainly be called the desktop mold DBMS from actuation of data on PC among the database management systems (it omits Following DBMS) which consist of database managers which operate data from keyboard equipment and a mouse. The desktop mold DBMS is equipped with the help system explaining the method and function of the use. There is a help system which can search with various means the help document which is created beforehand and stored on the hard disk drive unit as a well-known example about this help system. There are category retrieval, retrieval by keyword, a full-text search, a hyperlink, etc. in a retrieval means. Moreover, there is a function of "a help of condition dependence" which offers the interface jumped in the help document which corresponds according to the use condition of application.

[0003] The configuration of this conventional desktop mold DBMS and a help system is shown in drawing 1.

[0004] The help system in conventional DBMS is beforehand stored on the hard disk drive unit 106 by making the theory of a data base, and all the explanation documents of the immobilization about the function of DBMS into the help document 111, and displays a help document for the help document retrieval program 112 which searches the corresponding help document, and the help document 111 which it is as a result of retrieval on a display unit 100 by ejection and the help document display program 115 from a hard disk drive unit 106. The help system which offers a means to search like this well-known example to the help document which the DBMS vendor prepared beforehand will be called a static help system.

[0005] If "association" is searched as a keyword as an example of a static help system, in case it joins together for two or more relation first as a retrieval result of a help system, the help document which shows that there are internal linkage and an outer join as a class of association will be displayed. Then, it changes on the screen described in drawing 2, and it is called "internal linkage that the joint field of two relation is completely equal. The help document of the contents of immobilization [and] that an outer join shows that the record set with which the joint field corresponds is extracted from the record of one relation of all and another relation" is displayed. And as a still more detailed help document, when combining an order-received table and a customer table, the help document which explains the difference between an outer join and internal linkage using the specific example of immobilization may be displayed.

[0006] When "reference adjustment" is searched as a keyword as another example of a static help system, even if it performs registration and deletion of a record, it is a function to maintain the relation cypridium between tables in-reference adjustment."

"In order to realize reference adjustment, there are a thing of the level by declaration reference adjustment and a thing of the level by chain deletion / renewal of a chain. If it is with chain deletion / renewal of a chain and reference adjustment is realized, in declaration reference adjustment, it will become possible to perform the deletion actuation and update operation which cannot be performed. If a record is deleted or the value of the major key of the main table is changed, the contents of the association table will be changed and it will become possible to maintain reference adjustment."

The help document of the contents of the said immobilization is displayed.

[0007] The static help system is realized using the help document retrieval program 112 and the help document display program 115 which an operating system generally offers. By storing the explanation

document of application proper, such as DBMS and a spreadsheet, as a help document 111, the help system which performs a symbol description peculiar to application is realized. Creation of the help document in a static help system describes the tag and document which generally express the link and structure of a proper with a help system by RTF (Rich Text Format) etc., and is created by compiling with a help compiler.

[0008] In order to realize the development and actuation which a user means as another gestalt of a well-known example, there is a guidance device in which a suitable setup of the property of DBMS etc. is guided. This type of guidance device will be called a wizard. When a user wants to perform 02 and joint actuation of the "order-received" table 7a "customer" table seven a03 so that it may be shown in drawing 7 A as a wizard's example, there are some which choose association for which a screen will change to drawing 7 B and a user will ask it from three alternative if a mouse is double-clicked where connection seven a01 is chosen with the pointer of a mouse. This type of WIZADO is mounted independently with the static help system which an operating system offers. Unlike a static help system, a wizard does not only search / display the help document for which immobilization was prepared. "An order received" which is seven b01 described in drawing 7 B and the table name which the user defined like seven b03, and the "customer" who are seven b02 and the table name which the user defined like seven b04 may be extracted from the logical schema section 107, and it may be inserting and displaying into a document.

[Translation done.]

*** NOTICES ***

Japan Patent Office is not responsible for any damages caused by the use of this translation.

1.This document has been translated by computer. So the translation may not reflect the original precisely.

2.**** shows the word which can not be translated.

3.In the drawings, any words are not translated.

EFFECT OF THE INVENTION

[Effect of the Invention] As stated above, since a data base application developer can perform coding which was conscious only of the actuation to one table supposed by the configuration file definition, without being conscious of the complicated structure of data base table substance, according to this invention, he has the effect of data base application man day reduction.

[0079] Moreover, since no necessity for modification of the code of a program or recompilation is and it can respond only by Make Changes of data-base access a configuration file also when semantic attachment within the definition of data-base table substance and the application of the column data of a record changed into it, since the name with which a definition was supposed is used into the data-base access a configuration file of the program exterior by the column name for actuation in a program code, the effect of maintenance-cost reduction is.

[Translation done.]

* NOTICES *

Japan Patent Office is not responsible for any damages caused by the use of this translation.

1.This document has been translated by computer. So the translation may not reflect the original precisely.

2.***** shows the word which can not be translated.

3.In the drawings, any words are not translated.

TECHNICAL PROBLEM

[Problem(s) to be Solved by the Invention] The data access in the alias name which carried out the virtual table of the train name of a real table by performing the view definition of a data base with the above-mentioned conventional technology is possible.

[0011] However, in the data base application system which refers to one data base (307) from two or more applications like especially drawing 3 , when the form where an instruction is transmitted to a direct data base access interface (304,305,306) from an application program (301,302,303) is taken, the code of the application with which others are not changed, or the schema definition of a data base will be affected in the case of one or more application program modification.

[0012] When the code of application was changed, there is the necessity for recompilation, and big cost had to start, and it had to carry out by the application program having requested from the database manager modification of the data base schema definition including modification of a view definition of a database management system which is needed in the cases, such as an escape of a data base table, and modification within the application of the column data of a record the significance [modification] is given, and there was a problem that big cost started.

[0013] Moreover, when the SELECT sentence which defines a view includes the conditions of "association", Update operation only receives the train of Key-Preserved Table. The possibility of, Only when only one Key-Preserved Table exists in a view, deletion actuation The possibility of, Insertion actuation is possible only to the train of Key-Preserved Table. Although the real table with which each actuation is reflected has a limit of being one and a data base application developer can code in the reference actuation to a view, without being conscious of the structure of a real table when performing each actuation of updating, deletion, and insertion In each actuation of updating, deletion, and insertion, coating which was conscious of the structure of a real table had to be performed, and there was a problem that the consistency of actuation was missing.

[0014] This invention abolishes the necessity that he is conscious of the structure of a table also when performing updating and deletion / insertion actuation of a record at the time of application development. Reference actuation, It aims at giving consistency to the interface of updating and deletion / insertion actuation. Modification of the column data of the record furthermore used within application, It aims at abolishing the necessity for modification of the definition information in a database management system, and abolishing the necessity for code modification and recompilation of an application program in the case of modification and an escape of a table structure in the cases, such as modification column data gives the significance [modification].

[Translation done.]

*** NOTICES ***

Japan Patent Office is not responsible for any damages caused by the use of this translation.

1.This document has been translated by computer. So the translation may not reflect the original precisely.

2.**** shows the word which can not be translated.

3.In the drawings, any words are not translated.

MEANS

[Means for Solving the Problem] Since the above-mentioned technical problem is solved and the purpose of this invention is attained, the following means can be considered. A data base equipped with two or more real tables which are the sets of a real record which have one or more kinds of real column data, A database management means to acquire or change real column data of this data base, In a data base access system which consists of an access-control means to control acquisition or modification actuation of this real column data of said database management means, corresponding to a given SQL sentence Virtual table definition information which associates some real column data of one or more real tables of this data base as each virtual table column data of one virtual table, Virtual table operating instructions which acquire or change this virtual table column data are considered as an input. By generating one or more SQL sentences which operate it to real column data of one or more real tables of this data base, and giving this SQL sentence to this access-control means Acquisition or modification of real column data is operated. A result of acquisition of this real column data, or modification It is attained by offering a data base access system characterized by having the data-base-manipulation instruction transform-processing program outputted as acquisition or a modification result of virtual table column data.

[0016] Namely, this invention is set to data base application system using a relational database like drawing 4 . A data base application program using data stored in a relational database (401,402,403), Between data base access interface programs (404,405,406) which offer access in an SQL sentence to a Relational Database Management System, one or more tables of a relational database A virtual table column name and table column name contrast information on a configuration of a virtual table definition information table (604) shown in drawing 6 acquired from 502 of a configuration of being shown in virtual table definition file (409,411,413) drawing 5 (501) of the program exterior (601) And it changes into a virtual table based on a virtual table definition information table (414,415,416) which consists of joint column name set information (602) on a table, and record unique nature guarantee column name set information (603) in a table acquired from (503). Retrieval to this virtual table from an application program (401,402,403), Reflect a result of this actuation of insertion, modification, and an input of operating instructions of deletion to this virtual table. By changing into one or more SQL sentences which operate it to one or more tables, and publishing a generated SQL sentence to a data base access interface program (404,405,406) An instruction conversion program (408,410,412) which supposes a table structure as a virtual table to an application program (401,402,403) is arranged.

[0017] An instruction conversion program (408,410,412) receives virtual table record retrieval operating instructions. A virtual table column name train for retrieval, and information stored in a virtual table definition information table (414,415,416) from an input of retrieval conditions, From a virtual table column name which appears in input, a table name made applicable to retrieval is solved. A virtual table column name train for retrieval of an input, and a virtual table column name which appears in a retrieval condition It changes into a column name of a table. From virtual table definition information and a table name for retrieval A virtual table column name train for retrieval which generated a table connection condition and was changed into a column name of a table, Retrieval conditions changed into a column name of a table, a table name made applicable to retrieval, A retrieval instruction which generated a SELECT sentence of SQL and was generated from a table connection condition using a data base access interface of a relational database is published, and a column name of this retrieval result is changed and returned to a virtual table column name.

[0018] As opposed to virtual table record insertion operating instructions Moreover, information on a

virtual table definition information table, From an input of inserted-record information which consists of a virtual table column name and a pair of train of column data Divide this inserted-record information per table for each insertion, and each virtual table column name of this table unit inserted record is changed into a column name of a table. To a table which serves as a candidate of each insertion contrast, generate a SELECT sentence of SQL of an adjustment check and adjustment after virtual table record insertion is checked. When adjustment is maintained, an INSERT sentence of SQL which generated an INSERT sentence of SQL for every table, and was generated using a data base access interface of this relational database is published.

[0019] Moreover, a DELETE sentence of SQL which generated deletion conditions which can guarantee data consistency after deletion to be virtual table definition information from an input of deletion conditions about all groups of a table which constitutes a virtual table to virtual table record deletion operating instructions, generated a DELETE sentence of SQL for every table based on enumerated deletion conditions, and was generated using a data base access interface of this relational database is published.

[0020] Moreover, virtual table update record information which consists of a group of virtual table definition information, a virtual table column name, and virtual table column data to a virtual table record update operation instruction, Virtual table record retrieval actuation which makes all virtual table column names a virtual table all column name for retrieval, and makes updating conditions as a retrieval condition input from updating conditions is performed. Virtual table column data of virtual table update record information replaces data of a virtual table column contained in virtual table update record information on a retrieval result virtual table record, and said virtual table record insertion actuation is performed by considering this after [substitute] retrieval result virtual table record as an input.

[0021]

[Embodiment of the Invention] Hereafter, the gestalt of operation of this invention is explained to details.

[0022] Drawing 1 is the block diagram of the data base application system which applied this invention, drawing 2 is flow drawing showing the flow of the whole processing to the instruction from an application program, drawing 6 is the block diagram of a virtual table definition file, drawing 7 is one example of the table format of the data base made applicable [target] to actuation, and drawing 8 is one example of the virtual table definition file which constitutes a virtual table for the table format of drawing 7.

[0023] In drawing 1, a server machine (115) consists of CPU (102), a keyboard (103), external storage (104), a communication device (105), main storage (107), and a bus (106), stores a relational database (RDB) (120) in external storage (104), and loads and performs a Relational Database Management System (RDBMS) (122) on main storage (107).

[0024] Moreover, a client machine (116) consists of CPU (109), a keyboard (111), external storage (110), a communication device (108), main storage (114), and a bus (113), stores a virtual table definition file (121) on external storage (110), and loads and performs a data base application program (AP), (117), an instruction conversion program (123), and a virtual table setting program (119) on main storage (114). An instruction conversion program (123) reads the information on a virtual table definition file (121) into a virtual table definition information table (118) from external storage (110).

[0025] Virtual table definition information is stored in the virtual table definition information table (604) which has the structure shown by drawing 6 from the virtual table definition file (501) of a configuration of that drawing 5 shows. "The virtual table column name and table column name contrast information, and table joint column name set information (502)" on a virtual table definition file (501) are stored also in the joint column name information (602) on a table, when it stores in the virtual table column name and table column name contrast information (601) on a virtual table definition information table (604) and the column name of two or more tables is equivalent to one virtual table column name. Moreover (501) stores the record unique nature guarantee column name set information (503) in a table in the record unique nature guarantee column name set information (603) in a table on a virtual table definition information table (604).

[0026] In drawing 2, an instruction conversion program (202) reads into a virtual table definition information table (208) the information on the virtual table definition file (203) beforehand created using the virtual table setting program (204). If AP (201) publishes an instruction of retrieval actuation of a virtual table record, insertion actuation, deletion actuation, or update operation to an instruction conversion program (202) Based on this virtual table definition information, these virtual table record operating instructions The SELECT sentence of one or more SQL, Or it changes into the combination of an INSERT sentence or a DELETE sentence, publishes to the access interface (207) of the target relational database,

and is operated to RDB (206) through RDBMS (205).

[0027] The processing to which an instruction conversion program (202) performs hereafter the table format shown in drawing 7 to retrieval of a virtual table record, insertion, deletion, and an updating instruction for an example is explained.

[0028] Drawing 8 (801) is the example of the virtual table definition file for constituting a virtual table V (704) from a table A (701), a table B (702), and a table C (703), and drawing 9 (704) is a virtual table definition information table which reads and generates a virtual table definition file (801).

[0029] Drawing 10 is the block diagram of an instruction conversion program. The case where the virtual table column name train (1201) for retrieval and virtual table record retrieval conditions (1202) are hereafter published for the virtual table record retrieval instruction (1200) shown in drawing 12 as an input to the motion to virtual table record retrieval processing instruction of instruction conversion program instruction-conversion program (1020) for the virtual table V (704) is made into an example, and it explains.

[0030] Drawing 11 is flow drawing showing the processing to a retrieval instruction. In (step 1100), the virtual table definition information reading section (1016) reads a virtual table definition file (1017), and stores it in a virtual table definition information table (1015). At (step 1101), the instruction reception section (1100) receives the input of the virtual table column name train (1201) for retrieval, and virtual table record retrieval conditions (1202) as a retrieval instruction.

[0031] At (step 1102), the virtual table column name train (1201) for retrieval received at (step 1101) is passed to the virtual table column name processing section (1002) for retrieval, and virtual table record retrieval conditions (1202) are passed to the retrieval condition processing section (1003), respectively. The virtual table column name processing section (1002) for retrieval and the retrieval condition processing section (1003) register into the column name Management Department (1012) for retrieval the virtual table column name which appears during each input here. The virtual table column name registered in the input of (1200) here registers all the appearance virtual table column names that are the passages of (1203), and control moves from it to the SQL generation section (1001) immediately after the end.

[0032] At (step 1103), the SQL generation section (1001) requests solution of the table name for retrieval from the table name for retrieval, and the column name decision section (1004). In (step 1103), it is decided in the input of (1200) that it will be the table (1204) which serves as a candidate for retrieval from (1204).

[0033] At (step 1104), the SQL generation section calls the virtual table column name processing section (1002) for retrieval, and the virtual table record retrieval condition processing section (1003), and acquires the column name train for retrieval and retrieval conditions after column name conversion. At this time, the virtual table column name processing section (1002) for retrieval and the virtual table record retrieval condition processing section (1003) call the table name for retrieval, and the column name decision section (1004), acquire the table name made applicable to retrieval, and change a column name. It is made for the column name of a retrieval result to turn into a virtual table column name about the column name for retrieval here by surrounding with a double quotation mark after each column name for retrieval. the input of (1200) — receiving — each of (1206) — the output of the column name (1207) for retrieval and retrieval conditions (1208) is obtained.

[0034] At (step 1105), the SQL generation section calls the table connection condition generation section (1005), and acquires the connection condition about the table used as the candidate for retrieval. In the table connection condition generation section (1005), the table name for retrieval is acquired from the table name for retrieval, and the column name decision section (1004), and a table connection condition is generated based on the table joint column name information on a virtual table definition information table (1015). (1209) will be generated to the input of (1200).

[0035] At (step 1106), the SELECT sentence of SQL is generated based on the table name for retrieval generated at (step 1103), the column name for retrieval generated at (step 1104), retrieval conditions, and the table connection condition generated at (step 1105). To the input of (1200), if (1210), it is generated.

[0036] At (step 1107), the SQL sentence generated in the SQL instruction generation section (1001) in the SQL instruction issue section (1010) is published to a data base access interface (1018). At (step 1108), it changes into the format which uses a retrieval result with application again in the retrieval result processing section (1011).

[0037] Next, the case where a virtual table column name (1401) and virtual table column data (1402) are published for the virtual table record insertion instruction (1400) shown in drawing 14 as an input to the

motion to virtual table record insertion processing instruction of instruction conversion program instruction-conversion program (1020) for the virtual table V (704) is made into an example, and it explains.

[0038] Drawing 13 is flow drawing showing the processing to an insertion instruction. In (step 1300), the virtual table definition information reading section (1016) reads a virtual table definition file (1017), and stores it in a virtual table definition information table (1015). At (step 1301), the instruction reception section (1100) receives the input of the virtual table record insertion instruction (1400) which serves as a virtual table column name (1401) from virtual table column data (1402) as an insertion instruction.

[0039] At (step 1302), in the virtual table record division processing section (1006), a virtual table record is divided for every table corresponding to each column, and it registers with an inserted-record information management table (1013).

[0040] When a virtual table record insertion instruction (1400) is an input here, it becomes as shown in (1406) and they are the portion (1403) corresponding to Table A, a portion (1404) corresponding to Table B, and a portion (1405) corresponding to Table C, respectively. It sets up that it is unsettled about Table A, Table B, and Table C here.

[0041] A transaction is started at (step 1303). This is because virtual table record insertion actuation is offered as one actuation to the exterior of an instruction conversion program to consisting of one or more data variation orders on SQL level, so it is necessary to consider all data base modification processings in an instruction conversion program as one ATOMIKKU actuation in all.

[0042] At (step 1304), it investigates whether there is any table which is a candidate for data insertion and is not processed yet with reference to the inserted-record information management table (1013) in the insertion adjustment check section (1007). In the example of drawing 14, since Table A, Table B, and Table C are unsettled, it is YES.

[0043] At (step 1305), it is a table for insertion, and one unsettled table is chosen and it is referred to as T. In the example of drawing 14, it is arbitrary and Table A is set to T. At (step 1306), it inspects whether the record unique nature guarantee column in a table is shown in the table chosen at (step 1305) with reference to a virtual table definition information table. On the table A of an example, since ID column is a unique nature guarantee column, it is set to YES here.

[0044] It searches with (step 1309) whether the unique nature guarantee column of T exists in the inserted-record information about T altogether. In an example, the unique nature guarantee column of A is only ID, will exist altogether in the inserted-record information (1403) about A, and serves as YES. Since it becomes impossible to offer the adjustment guarantee at the time of inserting when it does not exist, the roll back of the transaction is carried out (step 1314), and it ends.

[0045] It searches with (step 1310) whether the data of the unique nature guarantee column of the inserted-record information about T and the record of equivalence exist. At this time, SQL used for inspection is a SELECT sentence with the conditional expression with which only the number of columns connected the conditional expression of the [column name = column data] of inserted-record information on AND conditions, and let the columns for retrieval be all columns. In the case of the example in drawing 14, checking SQL is set to (1407).

[0046] Since a retrieval result serves as a record whose number may not have been zero here, a record may be inserted (step 1307). Inserted-record information is compared with the record of a result searched with (step 1312) in (step 1311). Since it means that a mismatch occurs when it differs, the roll back of the transaction is carried out (step 1314), and it ends. When equivalent, it means that the same record already exists and it is not necessary to insert a record.

[0047] Since the record and inserted-record information on the result searched with the example are equivalent, it turns out that it is not necessary to insert a record in Table A. At (step 1308), T is set as a processing settled. It carries out about Table B and Table C like the following. Consequently, about Table B, 1409 and the SQL sentence of 1410 are generated about 1408 and Table C. After processing of Table B and Table C is completed, in branching of (step 1304), it is set to NO, and it commits (step 1313) and a transaction is ended.

[0048] Next, the case where the virtual table record deletion instruction (1700) shown in drawing 17 is published as a virtual table record deletion condition input to the motion to the virtual table record deletion instruction of instruction conversion program instruction-conversion program (1020) for the virtual table V (704) is made into an example, and it explains.

[0049] Drawing 15 is flow drawing showing the processing to a deletion instruction. In (step 1500), the virtual table definition information reading section (1016) reads a virtual table definition file (1017), and stores it in a virtual table definition information table (1015). At (step 1522), the instruction reception section (1100) receives the input of virtual table record deletion conditions as a virtual table record deletion instruction. In the example of drawing 17, (1700) is considered as a virtual table record deletion condition input.

[0050] At (step 1501), it divides for every table related in the virtual table record deletion conditions of an input in the virtual table record condition processing section (1008), and stores in a deletion condition information management table (1014). A deletion condition managed table (1014) consists of a deletion condition table (1600) with NxN structure, and a root flag table (1601) with 1xN structure here, when the number of tables contained in a virtual table is set to N, as shown in drawing 16.

[0051] In (step 1501), the conditions which divided the virtual table record deletion conditions of an input for every table are stored in the location of (T, T) of a deletion condition table (1600), when a corresponding table is set to T, respectively. A root flag table (1601) is initialized at OFF. (1701) shows the contents of the deletion condition table (1600) in a termination (step 1501) time to the input of (1700), and (1704) shows the contents of the root flag table.

[0052] A transaction is started at (step 1521). This is because virtual table record deletion actuation is offered as one actuation to the exterior of an instruction conversion program to consisting of one or more data variation orders on SQL level, so it is necessary to consider all data base modification processings in an instruction conversion program as one ATOMIKKU actuation in all.

[0053] At (step 1502), it inspects whether the table on which the element with which a root flag table (1601) corresponds serves as OFF exists. When it does not exist, it progresses to (step 1509). At (step 1503), a root flag chooses one of the arbitration from the tables of OFF, considers as the root, and sets the element of a corresponding root flag table (1601) to ON. Moreover, a current flag table is initialized at OFF.

[0054] At (step 1504), deletion conditions get down from close to the column with which the line corresponding to the root table of a deletion condition table (1600) corresponds, and it inspects whether the table whose current flag is OFF exists. It inspects whether the table on which conditions are not in close exists in the column with which one is chosen from them and current ** and a current flag are set when it exists (step 1505), and it is the candidate for joint of KARENTO, and the line corresponding to the root of a deletion condition table (1600) corresponds (step 1506).

[0055] When it does not exist (step 1504). When it exists, the table of the corresponding arbitration is set as a join, a search which makes the joint column of KARENTO and a join the train for retrieval a condition [the (current ** root) of a deletion condition table (1600)] is performed (step 1508), and it inspects whether a retrieval result exists (step 1519). When it does not exist, the roll back of the transaction is carried out (step 1520), and it ends.

[0056] When it exists, about a retrieval result record, the number of duplications is counted, respectively (step 1518), retrieval which makes retrieval conditions that with which only the number of record columns combined the conditions of "column name = column data" format by AND association based on the retrieval result record, and uses the column for retrieval as all columns is performed about each record, and the number of hits is counted (step 1517).

[0057] It inspects whether the record the number of duplications computed at (step 1518) here and whose number of hits called for at (step 1517) correspond exists (step 1515). When it does not exist, a deletion impossible flag is set as the join of a deletion condition table (1600), and the root (step 1516), and it progresses to (step 1506). When it exists, the conditions which generated the conditions with which only the number of record columns combined the conditions of "column name = column data" format by AND association about each record in agreement, and combined the each generated conditions by OR association are stored in (the join and the root) of a deletion condition table (1600) (step 1514), and it progresses to (step 1506).

[0058] (1702) is the contents of the deletion condition table constituted to the input of (1700) here. Moreover (1705), they are the contents of the root flag table in the time of (1702) being constituted. In (step 1502), when the table of OFF of the value of a root flag table does not exist, conditional expression of each line is published about each train of a deletion condition table (1600) to the table which joins together and corresponds by AND association (1513).

[0059] It is not operated about the table on which the deletion impossible flag is set up or close one is not [conditions] in the train to which a deletion condition table corresponds here (step 1512). (1703) is conditions of the deletion to each table generated to the input of (1700) here. It commits (step 1510) and a transaction is ended in the place which processing of all tables finished.

[0060] Next, the motion to the virtual table record update process instruction of an instruction conversion program is explained.

[0061] Drawing 18 is flow drawing showing the processing to an updating instruction, and drawing 19 is the example of the renewal instruction of a virtual table record. The virtual table definition information reading section (1016) reads a virtual table definition file (1017), and stores it in a virtual table definition information table (1015) (step 1800). The instruction reception section (1100) receives the input of the renewal instruction (1900) of a virtual table record which serves as a virtual table column name (1901) and virtual table column data (1902) from the renewal conditions (1903) of a virtual table record as an updating instruction (step 1801).

[0062] A transaction is started at (step 1810). This is because virtual table record update operation provides as one actuation to the exterior of an instruction conversion program to consisting of one or more data variation orders, so it needs to consider all data base modification processings in an instruction conversion program as one ATOMIKKU actuation in all on SQL level.

[0063] At (step 1802), all virtual table column names are specified as a virtual table record retrieval condition input as the renewal conditions of a virtual table record, and a virtual table column name for retrieval, and virtual table record retrieval processing is performed. A retrieval result virtual table record inspects one or more affairs of a certain thing (step 1803), as having no updating object record, in the case of zero affair, the roll back of the transaction is carried out (step 1808), and it is completed.

[0064] When a retrieval result exists, virtual table record deletion is performed as a virtual table record deletion condition input (step 1804). At (step 1805), it inspects whether a retrieval result set at (step 1802) is empty. When it is empty, it commits (step 1809) and a transaction is ended.

[0065] When it is not empty, the column data corresponding to the column name included in ejection and virtual table update record information in one of the arbitration is changed into the column data of virtual table update record information (step 1806), and virtual table record insertion actuation is performed as a virtual table record which inserts the changed virtual table record (step 1807), and it progresses to (step 1805).

[0066] According to the above-mentioned example, it is effective in becoming possible to change, without the executor of application changing the definition of a database management system for the object table of application.

[0067] In the above-mentioned example, the specification method of the target virtual table [conversion program / instruction] can consider the following methods.

[0068] Drawing 20 is the example of a configuration of the application with which application changes the target data at the assignment at the time of activation and which uses an instruction conversion program. External storage (2003) stores the execution file (2004) of the application which uses an instruction conversion program, a merchandise management virtual table definition file (2005), and a customer management virtual table definition file (2006).

[0069] When a goods data base application activation means (2001) is performed, (2004) is loaded to main storage (2007), stores the information on a merchandise management virtual table definition file (2005) in a merchandise management virtual table definition information table (2010), and becomes the goods data base application (2008) which operates a goods data control data base (2012).

[0070] When a customer database application activation means (2002) is performed, (2004) is loaded to main storage (2007), stores the information on a customer management virtual table definition file (2006) in a customer management virtual table definition information table (2011), and becomes the customer database application (2009) which operates a customer data control data base (2013).

[0071] In the above-mentioned example, it becomes possible for an application program to be able to change the target virtual table with an easy means to specify a definition file name at the time of starting, for example, to use the same application program of logic as another application like the goods data base application of the above-mentioned example, and customer database application.

[0072] Moreover, when [this] the virtual table operating instructions in the above-mentioned example may be SQL to a virtual table, the inside of the FROM phrase of a SELECT sentence and a DELETE

sentence and the INTO phrase of an INSERT sentence serves as a virtual table name.

[0073] In this example, even when SQL functor is extended, the effect by the virtual table can be realized.

[0074] For example, the example of the following escapes can be considered.

[0075] a SELECT news number, F1 (news, the 'report . text', 'report [the text {"Hitachi"}]'), and price FROM news table WHERE F — F1 and F2 are extended syntax 2 (news, 'report [the text {"Hitachi"}]') IS TRUE here.

[0076] In this case, the virtual table column name for retrieval in a previous example corresponds to NewsNo and F1(News, the 'report . text', 'report [the text {"Hitachi"}]') VALUE, virtual table record retrieval conditions correspond to F2(News, 'report [the text {"Hitachi"}]') IS TRUE, and the FROM phrase corresponding to the real table for retrieval and a connection condition are generated.

[0077] An output is SELECT NewsMaster.NewsNo [which is as follows], F1 (NewsMaster.News. 'the report . text', 'report [the text {"Hitachi"}]'), News Price.PriceFROM NewsMaster, and NewsPriceWHERE F2 (NewsMaster.News, 'report [the text {"Hitachi"}]') ISTRUE AND NewsMaster.NewsNo=NewsPrice.NewsNo.

[Translation done.]

* NOTICES *

Japan Patent Office is not responsible for any damages caused by the use of this translation.

1.This document has been translated by computer. So the translation may not reflect the original precisely.

2.**** shows the word which can not be translated.

3.In the drawings, any words are not translated.

DESCRIPTION OF DRAWINGS

[Brief Description of the Drawings]

[Drawing 1] It is the block diagram of the data base application system which applied this invention.

[Drawing 2] It is flow drawing showing the flow of the whole processing to the instruction from an application program.

[Drawing 3] It is the block diagram of the data base application system which refers to one data base from two or more applications.

[Drawing 4] It is the block diagram of the data base application system which applied this invention which refers to one data base from two or more applications.

[Drawing 5] It is the block diagram of a virtual table definition file.

[Drawing 6] It is the block diagram of a virtual table definition information table.

[Drawing 7] It is table format drawing of the data base used as a candidate for actuation with the gestalt of operation.

[Drawing 8] It is drawing showing one example of the virtual table definition file which constitutes a virtual table for the table format of drawing 7 .

[Drawing 9] It is drawing showing the virtual table definition information table which reads and generates the virtual table definition file of drawing 8 .

[Drawing 10] It is the block diagram showing the configuration of an instruction conversion program.

[Drawing 11] It is flow drawing showing the processing to a retrieval instruction.

[Drawing 12] It is flow drawing showing the example of the data about a retrieval instruction.

[Drawing 13] It is flow drawing showing the processing to an insertion instruction.

[Drawing 14] It is flow drawing showing the example of the data about an insertion instruction.

[Drawing 15] It is flow drawing showing the processing to a deletion instruction.

[Drawing 16] It is a block diagram showing the configuration of a deletion condition managed table.

[Drawing 17] It is the example of a virtual table record deletion instruction.

[Drawing 18] It is flow drawing showing the processing to an updating instruction.

[Drawing 19] It is the block diagram showing the example of the renewal instruction of a virtual table record.

[Drawing 20] It is drawing showing the example of a configuration of the application with which application uses the instruction conversion program which changes the target data by assignment at the time of activation.

[Description of Notations]

201 — Application program 202 — An instruction conversion program, 203 — Virtual table definition file, 204 — A virtual table setting program, 205 — Database management equipment, 206 — A data base, 207 — Access interface 208 — A virtual table definition information table, 1000 — Instruction reception section, 1001 — The SQL generation section, 1002 — The virtual table column name processing section for retrieval, 1003 — The virtual table record retrieval condition processing section, 1004 — The table name for retrieval, and the column name decision section, 1005 — The table connection condition generation section, 1006 — Virtual table record division processing section 1007 [— Deletion adjustment guarantee section,] — The insertion adjustment check section, 1008 — The virtual table record condition division processing section, 1009 1010 — The SQL instruction issue section, 1011 — Retrieval result processing section, 1012 — The virtual table for retrieval, 1013 — The column name Management Department, 1014 — Inserted-record information management table, 1015 — A virtual table definition

information table, 1016 — Virtual table definition information reading section 1017 [— [instruction conversion program.] — A virtual table definition file, 1018 — A data base access interface, 1019

[Translation done.]

**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ **BLACK BORDERS**
- ☐ **IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- ☒ **FADED TEXT OR DRAWING**
- ☐ **BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- ☐ **SKEWED/SLANTED IMAGES**
- ☐ **COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- ☐ **GRAY SCALE DOCUMENTS**
- ☐ **LINES OR MARKS ON ORIGINAL DOCUMENT**
- ☐ **REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- ☐ **OTHER:** _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.